



Universidad
Carlos III de Madrid

Departamento de Informática
Grado en Ingeniería Informática

PROYECTO FIN DE CARRERA

Asistente móvil para la realización de preparativos para la ITV de un automóvil

Autor: Fernando García Albaladejo

Tutor: Javier García Guzmán

Leganés, 5 septiembre de 2012

Título: ASISTENTE MÓVIL PARA LA REALIZACIÓN DE
PREPARATIVOS PARA LA ITV DE UN AUTOMÓVIL

Autor: Fernando García Albaladejo

Director: Javier García Guzmán

EL TRIBUNAL

Presidente: _____

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Se van a cumplir cuatro años de mi comienzo de la carrera y uno no puede dejar de pensar en todos los momentos que se han vivido en este largo y a la vez tan corto periodo. Ha habido momentos de duro trabajo y esfuerzo, momentos de relajación y, sobre todo, muchos momentos en los que tenías que dedicar largas tardes con los compañeros para sacar adelante prácticas.

Todos estos momentos indudablemente quedarán en el recuerdo para toda la vida, ya que éstos han servido para crecer como estudiante y como persona. Además, todos estos momentos no solo han servido para la adquisición de numerosos conocimientos, sino que también han servido para conocer a un grupo de compañeros, que no se les puede llamar simplemente compañeros, sino que son algo más, se les podría denominar sin temor a la confusión amigos, ya que siempre han estado ahí en los momentos que más se les necesitaba, y estoy seguro que esa unión y amistad que tengo con ellos se mantendrá para siempre. Especialmente quiero agradecer a mi compañero de prácticas, Nacho, todos los momentos que hemos dedicado a las mismas y todo lo positivo que me ha aportado, y cómo he dicho anteriormente, no me llevo un compañero, sino un amigo.

Por todo ello, esta no ha sido sólo una experiencia de aprendizaje a nivel de conocimiento, sino que ha servido para conocer a grandes personas.

También quiero hacer un especial agradecimiento a toda mi familia, a los que están y a los que se fueron, y en particular a mis padres y a mi hermana, sin los cuales todo esto hubiera sido evidentemente imposible, ya que ellos han sido los que han estado ahí día tras día, aguantando mis días buenos y mis días malos, y me animaban todos y cada uno de ellos para que todo fuera más fácil.

Tampoco me quiero olvidar de mis amigos de fuera de la universidad, tanto los de aquí de mi barrio a los que conozco desde que casi no tengo conocimiento, como todos los de mi preciado pueblo, a los que veo menos, pero que son cómo mis hermanos. Sé que sin todos ellos tampoco hubiera sido posible nada de esto y quiero agradecerles todo su apoyo y todos los momentos buenos y de risas que me han hecho pasar.

Quiero agradecer también todo el apoyo ofrecido por mi tutor, Javier García Guzmán. Él ha sido el que me ha ayudado a que el proyecto salga adelante aportándome sus conocimientos y numerosas ideas durante todo su desarrollo.

La finalización de este proyecto, marcará el fin de una de las etapas más importantes de mi vida, y dará comienzo a otra también muy importante, y por ello, a todas las personas nombradas anteriormente sólo puedo deciros una palabra, gracias.

Resumen

Las actuales aplicaciones móviles ofrecen a los usuarios diferentes servicios dependiendo de las necesidades que éstos busquen. Las opciones que se ofrecen con ellas son totalmente variadas y por ello cada día más personas hacen un mayor uso de ellas.

El problema que resuelve este proyecto consiste en las revisiones que un usuario debe realizar sobre su vehículo antes de realizarle la inspección técnica. Este problema, junto con el actual aumento del uso de dispositivos móviles, son las principales motivaciones de éste trabajo de fin de grado.

El principal objetivo de este trabajo es desarrollar una aplicación que permita revisar las partes más importantes de un vehículo para realizar la inspección técnica, así como la activación de alarmas que avisen de las próximas revisiones e informar de la documentación necesaria al usuario. Con ello no sólo se conseguirá que el usuario realice la inspección técnica de su vehículo con éxito, sino que también servirá para conseguir una mayor seguridad en el propio vehículo.

Todo ello se podrá realizar de una forma rápida e intuitiva. Lo único que será necesario es disponer de un dispositivo móvil adecuado para poder ejecutar la aplicación y tener instalada la misma en el dispositivo.

La metodología para el desarrollo de éste proyecto se ha basado en una metodología ágil de desarrollo denominada Scrum, ya que este tipo de metodologías se adaptan bastante bien al desarrollo de aplicaciones software. Estas metodologías se basan en la adaptación y el cambio durante el desarrollo de un proyecto.

El desarrollo del proyecto se ha basado en diferentes fases, que han servido para realizar una estructuración lógica del mismo.

Para el desarrollo de la aplicación se ha elegido la plataforma Android, ya que actualmente es la que mayor número de usuarios posee. Para este desarrollo se ha utilizado como lenguaje de programación Java, junto con ficheros XML y el entorno de desarrollo Eclipse. Cabe destacar, que para todas las funcionalidades se han realizado diferentes pruebas que garantizan el correcto funcionamiento de la aplicación.

El resultado que se ha obtenido ha sido una aplicación sencilla, amena e intuitiva, con la que se ofrecerá al usuario la posibilidad de realizar un seguimiento de su vehículo, facilitando de esta forma las inspecciones que se deben realizar sobre el mismo y que además al llevarlas a cabo se conseguirá una mayor seguridad también sobre el mismo.

Abstract

Today's mobile applications offer users different services depending on the needs to be searched. The options open to them are totally varied and therefore more and more people make more use of them.

The problem solved by this project is the reviews that a user must perform on your vehicle before performing technical inspection. These problems, along with the increased use of mobile devices today, are the main motivations for this end work to grade.

The main objective of this work is to develop an application for reviewing the most important parts of a vehicle to perform the technical inspection as well as the activation of alarms to alert and inform future revisions of the user documentation. These not only get the user to perform the technical inspection of your vehicle successfully, but will also serve to ensure greater security in the vehicle.

All this can be done in a fast and intuitive. All you will need is to have a suitable mobile device to run the application and have it installed on the device.

The methodology for the development of this project has been based on a methodology called Scrum agile development, as this type of methodologies adapted quite well to the development of software applications. These methodologies are based on adaptation and change in the course of a project.

Project development has been based on different phases, which have served to make a logical structuring it.

For the development of the application has been chosen the Android platform, which currently has the greatest number of users have. This development has been used as the programming language Java, along with XML files and Eclipse development The Area. Remarkably, for all the features different tests have been performed to ensure the proper functioning of the application.

The result that was obtained was a simple, pleasant and intuitive, which offer users the ability to track your vehicle, thus facilitating inspections to be performed on it and also to carry them out will ensure greater safety on it also.

Índice General de Contenidos

1	INTRODUCCIÓN	20
1.1	PROBLEMA	21
1.2	MOTIVACIÓN DEL PROYECTO	21
1.3	OBJETIVOS.....	22
1.4	METODOLOGÍA DE RESOLUCIÓN DEL PROBLEMA	23
1.5	TERMINOLOGÍA	23
1.6	CONTENIDO DE LA MEMORIA.....	24
2	ESTADO DEL ARTE	26
2.1	ANÁLISIS DE APLICACIONES MÓVILES SIMILARES	27
2.2	SISTEMA OPERATIVO SELECCIONADO.....	34
2.3	DESARROLLO ÁGIL COMO MÉTODO DE DESARROLLO DEL PROYECTO	43
2.4	CONCLUSIONES DEL ANÁLISIS.....	45
3	ESPECIFICACIÓN DE REQUISITOS	47
3.1	REQUISITOS DE USUARIO	48
3.2	CASOS DE USO	53

4	DISEÑO.....	67
4.1	DISEÑO EXTERNO.....	68
4.2	DISEÑO INTERNO	91
5	PRUEBAS DE LA APLICACIÓN	131
5.1	PRUEBAS DE ACEPTACIÓN	132
6	GESTION DEL PROYECTO	140
6.1	PLANIFICACIÓN DEL PROYECTO	141
6.2	HERRAMIENTAS UTILIZADAS	146
6.3	ANÁLISIS ECONÓMICO	147
7	CONCLUSIONES	151
7.1	CONCLUSIONES.....	152
7.2	LÍNEAS FUTURAS.....	154
	BIBLIOGRAFIA	155
	APÉNDICE A. MANUAL DE USUARIO	158

Índice de Figuras

Figura 1: Itv pantalla 2.....	29
Figura 2: Itv pantalla 1.....	29
Figura 3: Acar pantalla 1.....	30
Figura 4: Acar pantalla 2.....	30
Figura 5: 05 min pantalla 1.....	31
Figura 6: 05 min pantalla 2.....	31
Figura 7: Libro de mantenimiento pantalla 1.....	32
Figura 8: Libro de mantenimiento pantalla 2.....	32
Figura 9: Porcentaje de uso de Sistemas operativos para móviles.....	35
Figura 10: Uso diferentes versiones Android.....	37
Figura 11: Arquitectura de Android.....	38
Figura 12: Ciclo de vida actividades en Android.....	41
Figura 13: Diagrama metodología Scrum.....	45
Figura 14: Casos de Uso.....	54
Figura 15 - Pantalla inicial primer prototipo.....	69
Figura 16 - Pantalla que revisar primer prototipo.....	70
Figura 17 - Pantalla revisar frenos primer prototipo.....	71
Figura 18 - Pantalla documentación necesaria primer prototipo.....	72
Figura 19 - Pantalla próxima revisión primer prototipo.....	73
Figura 20 - Pantalla activar recordatorio primer prototipo.....	74
Figura 21 - Pantalla recordatorios primer prototipo.....	75
Figura 22: Pantalla inicial.....	76
Figura 23: Pantalla Que revisar.....	78
Figura 24: Pantalla lista Alumbrado.....	78
Figura 25: Pantalla primera alumbrado.....	80
Figura 26: Pantalla segunda alumbrado.....	80
Figura 27: Pantalla lista modificada alumbrado.....	81
Figura 28: Pantalla próxima revisión.....	82
Figura 29: Pantalla información periodos inspección.....	83
Figura 30: Pantalla activar notificación.....	84
Figura 31: Pantalla diálogo confirmar notificación.....	85
Figura 32: Pantalla ver notificaciones.....	87
Figura 33: Pantalla ver documentación 1.....	88
Figura 34: Pantalla ver documentación 2.....	89
Figura 35: Pantalla ver documentación 3.....	90
Figura 36: Diagrama de componentes de la aplicación.....	93
Figura 37: Diagrama de secuencia revisar alumbrado coche.....	95
Figura 38: Diagrama de secuencia revisar neumáticos coche.....	96
Figura 39: Diagrama de secuencia revisar frenos coche.....	97

Figura 40: Diagrama de secuencia revisar general coche	98
Figura 41: Diagrama de secuencia revisar alumbrado motocicleta	99
Figura 42: Diagrama de secuencia revisar neumáticos motocicleta.....	100
Figura 43: Diagrama de secuencia revisar frenos motocicleta	101
Figura 44: Diagrama de secuencia revisar general motocicleta.....	102
Figura 45: Diagrama de secuencia documentación y notificaciones	104
Figura 46: Diagrama de clases actividad principal	106
Figura 47: Diagrama de clases actionBar	107
Figura 48: Diagrama de clases que revisar coche	108
Figura 49: Diagrama de clases que revisar motocicleta.....	109
Figura 50: Diagrama de clases revisar alumbrado coche.....	110
Figura 51: Diagrama de clases revisar frenos coche	111
Figura 52: Diagrama de clases revisar neumáticos coche.....	112
Figura 53: Diagrama de clases revisar general coche	113
Figura 54: Diagrama de clases revisar alumbrado motocicleta	114
Figura 55: Diagrama de clases revisar frenos motocicleta.....	115
Figura 56: Diagrama de clases revisar neumáticos motocicleta	116
Figura 57: Diagrama de clases revisar general motocicleta.....	117
Figura 58: Diagrama de clases notificaciones y documentación	119
Figura 59: Diagrama de clases documentación.....	120
Figura 60: Diagrama de clases base de datos	121
Figura 61: Creación de la base de datos.....	122
Figura 62: Método insert.....	123
Figura 63: Método query	124
Figura 64: Método delete.....	125
Figura 65: Método update	126
Figura 66: Estructura base de datos.....	126
Figura 67: Estructura base de datos 2.....	127
Figura 68: Ejecución base de datos	127
Figura 69: Modelo entidad-relación de los datos	128
Figura 70: Diagrama de Gantt planificación inicial.....	143
Figura 71: Diagrama de Gantt final	145
Figura 72: Aplicación final pantalla inicial	159
Figura 73: Aplicación final seleccionar tipo vehículo a revisar.....	160
Figura 74: Aplicación final que parte revisar coche	161
Figura 75: Aplicación final partes alumbrado coche	162
Figura 76: Aplicación final luces de Posición.....	163
Figura 77: Aplicación final luces Posición 2.....	164
Figura 78: Aplicación final lista alumbrado 2	165
Figura 79: Aplicación final próxima revisión	166
Figura 80: Aplicación final periodo revisión vehículos.....	167

Figura 81: Aplicación final próxima fecha revisión.....	168
Figura 82: Aplicación final activar notificación	169
Figura 83: Aplicación final notificaciones almacenadas.....	170
Figura 84: Aplicación final confirmar notificación.....	171
Figura 85: Aplicación final documentación necesaria.....	172

Índice de tablas

Tabla 1: Comparativa de aplicaciones.....	34
Tabla 2: Versiones Android	36
Tabla 3: Formato tabla requisitos de usuario	48
Tabla 4: RF - 001 - Qué revisar en el vehículo	49
Tabla 5: RF - 002 - Próxima revisión del vehículo.....	49
Tabla 6: RF - 003 - Ver Notificaciones	49
Tabla 7: RF - 004 - Ver documentación necesaria para la inspección.....	49
Tabla 8: RF - 005 - Revisar alumbrado.....	50
Tabla 9: RF - 006 - Revisar neumático	50
Tabla 10: RF - 007 - Revisar Frenos	50
Tabla 11: RF - 008 - Revisión general del vehículo	50
Tabla 12: RF - 009 - Selección del tipo de vehículo	51
Tabla 13: RF - 010 - Seleccionar fecha de matriculación vehículo	51
Tabla 14: RF - 011 - Mostrar periodos de las inspecciones	51
Tabla 15: RF - 012 - Añadir notificación	51
Tabla 16: RF - 013 – Insertar matricula vehículo.....	52
Tabla 17: RF - 014 - Elegir fecha notificación	52
Tabla 18: RF - 015 - Eliminar notificaciones	52
Tabla 19: RF - 017 - Retornar a la pantalla de inicio.....	52
Tabla 20: RR - 001 - Versión del Sistema Operativo.....	53
Tabla 21: RR - 002 - Versión del SDK de Java	53
Tabla 22: CU - 01 - Revisar partes vehículo	55
Tabla 23: CU-02 – Revisar las diferentes partes del alumbrado coche.....	56
Tabla 24: CU-03 - Revisar las diferentes partes de los frenos coche	56
Tabla 25: CU-04 - Revisar las diferentes partes de los neumáticos coche.....	57
Tabla 26: CU-05 - Revisar las diferentes partes coche en general.....	57
Tabla 27: CU – 06 - Revisar partes alumbrado motocicleta	58
Tabla 28: CU- 07 - Revisar partes freno motocicleta.....	59
Tabla 29: CU- 08 - Revisar partes neumáticos motocicleta	60
Tabla 30: CU- 09 - Revisar partes general motocicleta	61
Tabla 31: Cu-10 - Calcula de fecha próxima inspección	62
Tabla 32: CU-11 - Activar notificación.....	63
Tabla 33: CU-12 - Ver periodos inspección vehículos	64
Tabla 34: CU-13 - Ver notificaciones almacenadas	64
Tabla 35: CU-14 - Eliminar notificaciones	65
Tabla 36: CU-15 - Ver documentación necesaria	65

Tabla 37: Formato tabla pruebas de aceptación	132
Tabla 38: PA - 01 - Inicio correcto aplicación	133
Tabla 39: PA - 02 - Cargar lista alumbrado.....	133
Tabla 40: PA - 04 - Cargar lista frenos	134
Tabla 41: PA - 04 - Cargar lista neumáticos.....	134
Tabla 42: Cargar lista generales	135
Tabla 43: PA - 06 - Botón "home"	135
Tabla 44: PA - 07 - Lista periodos	136
Tabla 45: PA - 08 - Cálculo fecha inspección	136
Tabla 46: PA - 09 - Activar notificación	137
Tabla 47: PA - 10 - Mostrar notificaciones	137
Tabla 48: PA - 11 - Eliminar notificaciones.....	137
Tabla 49: PA - 12 - Transición correcta actividades	138
Tabla 50: Imágenes correctas aplicación	138
Tabla 51: Hardware ordenador sobremesa	146
Tabla 52: Hardware smartphone Wildfire.....	146
Tabla 53: Hardware smartphone Huawei	146
Tabla 54: Software utilizado.....	147
Tabla 55: Costes personal	148
Tabla 56: Costes herramientas hardware	149
Tabla 57: Costes herramientas software	149
Tabla 58: Costes indirectos	150
Tabla 59: Costes totales	150

1 INTRODUCCIÓN

1	INTRODUCCIÓN	20
1.1	PROBLEMA	21
1.2	MOTIVACIÓN DEL PROYECTO	21
1.3	OBJETIVOS.....	22
1.4	METODOLOGÍA DE RESOLUCIÓN DEL PROBLEMA.....	23
1.5	TERMINOLOGÍA	23
1.5.1	GLOSARIO DE TÉRMINOS.....	24
1.5.2	ABREVIATURAS	24
1.6	CONTENIDO DE LA MEMORIA.....	24

En este capítulo se realizará una pequeña introducción del presente proyecto, en el cual se expondrán el principal problema existente, las motivaciones y los objetivos del mismo, la metodología utilizada y los contenidos de la presente memoria.

1.1 PROBLEMA

El principal problema que se desea resolver con este proyecto radica en los inconvenientes y dificultades que encuentran algunos usuarios a la hora de realizar la inspección técnica de sus vehículos.

Estas inspecciones son de vital importancia, ya que con ellas se regula la seguridad vial que los vehículos ofrecen a sus usuarios, y por lo tanto, con esta aplicación no sólo se ayudará al usuario que la utilice, sino que de forma indirecta, también ayudará a los demás usuarios de vehículos, ya que la seguridad de un vehículo puede involucrar la de otros.

En muchas ocasiones la de dificultad que encuentran los usuarios es que no saben exactamente que revisar en su vehículo antes de realizar dichas inspecciones o no conocen la documentación que es necesaria presentar para poder pasar la misma con éxito.

En otros casos el problema de algunos usuarios es que no saben con exactitud el periodo con el que debe realizar la inspección de su vehículo o incluso se le puede llegar a olvidar la fecha en la que deben llevarla a cabo.

Este problema se suele presentar normalmente en usuarios nuevos y que por lo tanto son poco experimentados en estas situaciones.

Por todo ello, esta aplicación que se va a desarrollar, será de gran utilidad para muchos usuarios que no saben con exactitud cómo o cuando deben afrontar la inspección de su vehículo y, por lo tanto, en muchas ocasiones servirá para pasarla con éxito y ahorrar por lo tanto dinero y tiempo.

Este problema se puede resolver mediante el desarrollo de dicha aplicación y usándola en los nuevos dispositivos móviles, que hacen que esto se pueda realizar de una forma cómoda y en cualquier lugar y momento que se desee.

1.2 MOTIVACIÓN DEL PROYECTO

La importancia que se le está atribuyendo a la aparición de las nuevas tecnologías móviles, unido al aumento del número de personas que cada día dispone

de un dispositivo móvil inteligente y a la gran dependencia que éstos están generando cada vez más fuertemente sobre la sociedad, son las principales motivaciones para llevar a cabo el desarrollo de este trabajo fin de grado.

La principal motivación es conseguir resolver el problema que se comentó en el punto anterior, ya que se cree que puede ser una aplicación de gran beneficio para los usuarios que hagan uso de ella, previniendo de este modo que dichos usuarios tengan que asistir más de una vez a la inspección de su vehículo porque haya habido algún fallo por no haber sido revisado cualquier punto con anterioridad. Además, también es novedosa la incorporación de recordatorios para que los usuarios no se olviden de la fecha en la que tienen que revisar sus vehículos.

Por todo ello, existe un gran afán por desarrollar una aplicación que pueda servir para mejorar y facilitar situaciones que se producen diariamente y que pueda ayudar a sus usuarios para ahorrar tiempo e incluso dinero, de una manera fácil y sencilla, y que cada día esté al alcance de más personas. Esta aplicación tendrá como principal enfoque los vehículos de los usuarios, ya que será sobre ellos sobre los que se lleven a cabo las revisiones antes de llevar a cabo las inspecciones.

1.3 OBJETIVOS

El principal objetivo de este proyecto **es el desarrollo de una aplicación para dispositivos móviles basados en el sistema operativo Android, que permita a sus usuarios realizar una serie de preparativos sobre sus vehículos para poder pasar la inspección técnica (ITV) de los mismos con éxito.**

Además de este objetivo principal, este proyecto presenta otros objetivos específicos que se encuentran enumerados a continuación:

- La aplicación deberá proporcionar información al usuario sobre las partes del vehículo que éste deberá revisar para poder pasar satisfactoriamente la inspección técnica de dicho vehículo.
- La aplicación ofrecerá información al usuario sobre los periodos en los que cada tipo de vehículo deberá realizar la revisión técnica del mismo.
- El usuario podrá calcular la fecha en la que deberá realizar la próxima inspección técnica de su vehículo, a partir de la fecha de matriculación del mismo, y, además se le ofrecerá la opción de poder activar una notificación que le servirá de recordatorio días antes de la propia inspección.
- El usuario, además, también tendrá la opción de eliminar dichas alarmas en el momento que lo crea preciso.

- La aplicación proporcionará información al usuario sobre toda la documentación que será necesaria para poder pasar con éxito la inspección técnica de su vehículo.

1.4 METODOLOGÍA DE RESOLUCIÓN DEL PROBLEMA

El proceso para la resolución del problema se ha resuelto mediante un método incremental, puesto que inicialmente no se tenían totalmente claros los requisitos que la propia aplicación debía cumplir. Por ello, éstos se han ido modificando a lo largo del desarrollo del proyecto, en otros casos se han eliminado otros que se ha considerado necesario y en otras ocasiones se han añadido nuevos que se pensaban fundamentales para la obtención de la aplicación final.

Este modo incremental que se planteó desde el inicio del proyecto se basa en alguno que se propone en algunas de las metodologías ágiles. Estas metodologías se han basado principalmente en las reuniones que se han mantenido durante el desarrollo del proyecto y en las cuales se iban negociando los cambios que se debían llevar a cabo en los requisitos para la siguiente reunión. En dichas reuniones se evaluaban los prototipos de reuniones anteriores y se especificaban como debían ser para las siguientes. Por lo tanto el cliente presenta un papel vital en dicha metodología.

Además en las propias reuniones también se marcaba el tiempo en el que se debían cumplir cada una de las tareas deseadas. Para llevar a cabo el control del tiempo y las tareas que se han ido realizando durante el desarrollo del proyecto se ha hecho uso de la herramienta *Redmine*, en la cual es posible almacenar cada una de las diferentes tareas, almacenar el tiempo y autor que de cada una de ellas e indicar en cada momento el porcentaje de desarrollo alcanzado hasta llegar a su finalización.

1.5 TERMINOLOGÍA

En este apartado se detalla un glosario en el que se explican brevemente algunos términos que se han usado en el presente documento con el fin de que el lector pueda comprender el total del contenido.

1.5.1 GLOSARIO DE TÉRMINOS

App es la forma de denominar a las aplicaciones informáticas para dispositivos móviles como son los Smartphone o tabletas.

iOS es el sistema operativo móvil de Apple desarrollado inicialmente para iPhone y utilizado más tarde para dispositivos como iPod, iPad o Apple TV.

Android es el sistema operativo basado en Linux que está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

Smartphone es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional.

1.5.2 ABREVIATURAS

API del inglés Application Programming Interface, Interfaz de programación de aplicaciones. Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

SDK del inglés Software Development Kit, Kit de desarrollo de software. Es un conjunto de bibliotecas que ofrecen al desarrollador unas funcionalidades básicas que permiten crear programas con mayor facilidad.

XML eXtended Markup Language. Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

ITV proveniente de Inspección Técnica del Vehículo. Es el lugar donde los vehículos realizarán las inspecciones correspondientes cada cierto periodo de tiempo.

1.6 CONTENIDO DE LA MEMORIA

En este apartado se detallará brevemente la estructura de la presente memoria:

- **Capítulo 1, Introducción.** Este primer capítulo servirá para exponer la motivación por la que se ha llevado a cabo el proyecto, así como los objetivos que se persiguen con el.

- Capítulo 2, **Estado del Arte**. En este capítulo se realizará un análisis sobre aplicaciones que se encuentran en el mercado y que cumplen objetivos similares a los que se persiguen y que por lo tanto podrán servir de apoyo en cierta medida para el desarrollo de la aplicación deseada. Además, en este mismo capítulo, también se desarrollarán los casos de uso necesarios, así como los requisitos de usuario y el plan de pruebas de la aplicación.
- Capítulo 3, **Especificación de requisitos**. En este capítulo se presentarán los requisitos que debe cumplir el sistema, así como los casos de uso y el plan de pruebas correspondiente.
- Capítulo 4, **Diseño**. Este capítulo servirá para presentar el diseño de la aplicación a través de los prototipos de la misma, el diagrama de clases y algunos diagramas de secuencia.
- Capítulo 5, **Pruebas de la aplicación**. En este capítulo se expondrá el resultado de todas las pruebas que garantizarán que la aplicación funciona de forma correcta.
- Capítulo 6, **Gestión del Proyecto**. En este quinto capítulo se llevará a cabo una gestión del proyecto en la cual se especificarán puntos como la planificación y los costes económicos resultantes del mismo.
- Capítulo 7, **Conclusiones**. En este último capítulo se presentarán las conclusiones, tanto personales como del proyecto, a las que se ha llegado durante el desarrollo del mismo.
- Anexo A, **Manual de usuario**: Este apartado servirá para mostrar el manual que debe seguir el usuario para hacer un uso adecuado de la aplicación.
- **Bibliografía**: Apartado que servirá para incluir todas las fuentes de información utilizadas.

2 ESTADO DEL ARTE

2	ESTADO DEL ARTE	26
2.1	ANÁLISIS DE APLICACIONES MÓVILES SIMILARES	27
2.1.1	ITV.....	28
2.1.2	ACAR	29
2.1.3	05 MIN: MANTENIMIENTO DE SU VEHICULO	31
2.1.4	LIBRO MANTENIMIENTO	32
2.2	SISTEMA OPERATIVO SELECCIONADO.....	34
2.2.1	SISTEMAS OPERATIVOS EN EL MERCADO.....	35
2.2.2	VERSIONES EN ANDROID	35
2.2.3	ARQUITECTURA DEL SISTEMA OPERATIVO ANDROID	37
2.2.4	CICLO DE VIDA DE LAS ACTIVIDADES EN ANDROID	40
2.2.5	ENTORNO DE DESARROLLO INTEGRADO.....	42
2.3	DESARROLLO ÁGIL COMO MÉTODO DE DESARROLLO DEL PROYECTO	43
2.3.1	METODOLOGIA AGIL SELECCIONADA	43
2.4	CONCLUSIONES DEL ANÁLISIS.....	45

El presente capítulo servirá para realizar un análisis sobre la aplicación que se va a desarrollar, y por lo tanto, servirá de base para el diseño que se realizará seguidamente de la misma.

Dicho análisis constará de una serie de pautas que serán vistas en los siguientes puntos.

2.1 ANÁLISIS DE APLICACIONES MÓVILES SIMILARES

Para llevar a cabo este apartado es importante tener claros los objetivos que se pretenden conseguir con este proyecto. Dichos objetivos son los que quedan detallados en el apartado 1.2 de la presente memoria.

Teniendo en cuenta dichos objetivos, se ha llevado a cabo una búsqueda de diferentes aplicaciones que presentaran funcionalidades parecidas o iguales a las que se pretenden conseguir. Con esto se pretende obtener una visión sobre las diferentes aplicaciones que existen en el mercado actualmente y sus diversas funcionalidades, y las ideas que pueden aportar éstas para el desarrollo de la aplicación deseada.

La búsqueda que se han realizado para el estudio de aplicaciones similares se ha llevado a cabo mediante tres medios. El primer medio se ha basado en la búsqueda a través de la Web, y para ello se ha hecho uso de un navegador común como es Google. Por otro lado, se ha hecho uso de la herramienta Google Play, en la cual se pueden encontrar, entre otras cosas, aplicaciones basadas para los dispositivos Android. De manera similar a la anterior, se ha hecho uso de la herramienta App Store, en la cual se pueden encontrar aplicaciones pero en este caso para la plataforma móvil basada en iOS.

En el caso del buscador Web Google, las búsquedas han ido orientadas a aplicaciones que estuvieran relacionadas con las inspecciones de vehículos. Para ello se han utilizado términos de búsqueda como los que se pueden ver a continuación:

- ITV para móviles.
- Aplicación para móvil ITV.

Por otro lado, en el caso de las herramientas que proporciona cada uno de las plataformas móviles comentadas anteriormente se han realizado las búsquedas de forma algo diferente, ya que estas herramientas ya cuentan con secciones que clasifican los contenidos por categorías, y por lo tanto, en este caso la categoría que se ha seleccionado ha sido la de aplicaciones y una vez dentro los términos de búsqueda que se han seguido han sido los siguientes:

- ITV.

- Revisión vehículo.

Todas estas búsquedas fueron llevadas a cabo durante la semana del 25 al 29 de enero de 2012. En algunas de las aplicaciones analizadas se han tenido en cuenta la actualización de versiones que han sufrido.

Por otro lado, también se obtuvieron otros resultados durante las búsquedas, pero éstos fueron descartados ya que se pensaba que no aportaban ideas novedosas y atractivas a la aplicación que se deseaba desarrollar, y por lo tanto fueron desechados.

A continuación se hará un análisis sobre las aplicaciones que se han encontrado y que guardan más relevancia con la aplicación que se desea desarrollar.

2.1.1 ITV

Se trata de una aplicación basada en el sistema operativo de Android y denominada *ITV* [6]. Entre las funcionalidades de esta aplicación podemos encontrar las siguientes:

- Informar al usuario de la frecuencia con la que tiene que realizar la revisión a su automóvil.
- Cálculo de equivalencia de neumáticos para comprobar que cumplen con los requisitos obligatorios.
- Proveer al usuario de un manual en el que se explique el procedimiento de las Inspecciones técnicas.
- Proveer al usuario de un manual en el que se detallen las reformas que se pueden realizar sobre un automóvil.

Esta aplicación, Figura 2 y Figura 1, está orientada a todos aquellos usuarios que vayan a realizar una inspección técnica de su vehículo.

Por otro lado, las funcionalidades que se encargan de informar al usuario sobre el procedimiento de las inspecciones y de las reformas que se pueden realizar sobre un automóvil pueden resultar algo incómodas para dicho usuario, ya que para informarle lo que hace la aplicación es descargar un manual, en formato “pdf”, de la página del ministerio de industria, energía y turismo del gobierno de España. Esto puede resultar incómodo para el usuario ya que se trata de manuales de una extensa longitud y en muchas ocasiones el usuario no dispondrá del tiempo suficiente para poder leerlo entero.



Figura 2: Itv pantalla 1



Figura 1: Itv pantalla 2

2.1.2 ACAR

Al igual que en el anterior caso también se trata de una aplicación [7] desarrollada para el sistema operativo Android. Entre sus funcionalidades se encuentran muchas que no son de interés para el desarrollo de la aplicación, por ello se acotará la descripción de dichas funcionalidades centrándose en las que verdaderamente puedan aportar sugerencias. Estas funcionalidades son las siguientes:

- Mostrar al usuario la duración, tanto en kilómetros, como en tiempo, de una gran selección de piezas de las que está formado el automóvil.
- Ofrecer la posibilidad al usuario de activar alarmas que sirvan para alertarle de que tiene que proceder a cambiar alguna de las piezas.
- Ofrecer la posibilidad al usuario de modificar los kilómetros y el tiempo de duración que tiene cada una de las piezas, e incluso añadir nuevas.

Esta aplicación, Figura 3 y Figura 4 está enfocada para todos aquellos usuarios que posean un automóvil y deseen llevar un mantenimiento exhaustivo de su automóvil.

La funcionalidad que más llama la atención de esta aplicación son los recordatorios que se pueden programar para avisar de futuras operaciones de mantenimiento, lo cual resulta de gran utilidad, ya que el usuario será avisado cada vez que una pieza necesita una revisión.

Como característica negativa decir que la aplicación hace uso de una gran cantidad de datos del automóvil, lo cual puede hacer que usuarios inexpertos no le saquen su máximo rendimiento.



Figura 3: Acar pantalla 1



Figura 4: Acar pantalla 2

2.1.3 05 MIN: MANTENIMIENTO DE SU VEHICULO

En este caso se trata de una aplicación desarrollada para el sistema operativo iOS.

Se trata de una aplicación de consulta rápida, que se encarga de explicar al usuario de forma breve y concisa los pasos que éste debe seguir para realizar un buen mantenimiento de su automóvil. Todo esto lo hace de una forma clara, ordenada y rápida, facilitando así la tarea al usuario. En concreto muestra como realizar el mantenimiento del alumbrado, neumáticos, niveles de filtros y frenos del automóvil.

La característica más atractiva de esta aplicación, Figura 5 y Figura 6 es que muestra la información de una forma muy resumida, lo cual resulta muy cómodo para el usuario, ya que de esta forma obtendrá la información de lo verdaderamente necesario para el mantenimiento de su automóvil de forma rápida.

Como punto negativo, decir que esta aplicación no se centra concretamente en los preparativos de la ITV, pero puede servir como referencia para la realización de un buen mantenimiento del automóvil, lo cual será de vital importancia para pasar con éxito dicha revisión.



Figura 5: 05 min pantalla 1



Figura 6: 05 min pantalla 2

2.1.4 LIBRO MANTENIMIENTO

Al igual que la anterior aplicación, esta también se encuentra desarrollada para el sistema operativo iOS.

Se trata de una aplicación similar a la comentada anteriormente, “aCar”, desarrollada por Android, ya que se puede configurar la frecuencia de revisión de diferentes piezas que forman parte del automóvil, tanto en kilómetros como en tiempo. Con esta aplicación se podrá llevar un control del estado del automóvil y así el usuario no se olvidará de cuando tiene que pasar la siguiente revisión, ya que ofrece la posibilidad de añadir recordatorios.

Una de las características más atractivas de esta aplicación, Figura 7 y Figura 8, es que se encuentra desarrollada en varios idiomas, en concreto en 5, entre ellos el Español.

Al igual que la anterior aplicación comentada, también será necesario configurar la frecuencia de revisión de cada una de las piezas, ya que éstas vendrán con una frecuencia por defecto que posiblemente no sea la correcta para cada marca. Esto puede resultar tedioso para los usuarios, ya que son muchos datos los que debe introducir.



Figura 7: Libro de mantenimiento pantalla 1



Figura 8: Libro de mantenimiento pantalla 2

Tras realizar un análisis de las diferentes aplicaciones encontradas se puede obtener una tabla en la que se refleja si cada una de ellas cumple con los objetivos que se persiguen para el desarrollo de nuestra aplicación. Con esto se conseguirá ver si alguna de las aplicaciones observadas podría suplir las funcionalidades que se persiguen para la aplicación.

- La aplicación deberá proporcionar información al usuario sobre las partes del vehículo que éste deberá revisar para poder pasar satisfactoriamente la inspección técnica de dicho vehículo.
- La aplicación ofrecerá información al usuario sobre los periodos en los que cada tipo de vehículo deberá realizar la revisión técnica del mismo.
- El usuario podrá calcular la fecha en la que deberá realizar la próxima inspección técnica de su vehículo a partir de la fecha de matriculación del mismo, y además se le ofrecerá la opción de poder activar una notificación que le servirá de recordatorio días antes de la propia inspección.
- El usuario además también tendrá la opción de eliminar dichas alarmas en el momento que lo crea preciso.
- La aplicación proporcionará información al usuario sobre toda la documentación que será necesaria para poder pasar con éxito la inspección técnica de su vehículo.

Aplicación Objetivo	ITV	ACAR	05 MIN: MANTENIMIENTO DE SU VEHICULO	LIBRO MANTENIMIENTO
Ofrecer Información partes vehículo	NO	SI	SI	SI
Información periodos	SI	NO	NO	NO
Calculo próxima inspección	NO	NO	NO	NO
Activar Notificación	NO	SI	NO	SI
Información documentación	SI	NO	NO	NO
Baja complejidad	SI	SI	SI	SI

Tabla 1: Comparativa de aplicaciones

Tras el análisis realizado de las aplicaciones similares que se pueden encontrar en el mercado, se puede observar que ninguna de ellas cumple con todas las funcionalidades que se desean para la aplicación a desarrollar.

La Tabla 1 ofrece una clara visión de las funcionalidades que cumple cada una de ellas. Cabe destacar que ninguna de ellas ofrece la posibilidad de calcular la próxima fecha para la inspección del vehículo. También cabe destacar que, aunque se marquen dos aplicaciones con la opción de poder activar notificaciones, ninguna de ellas lo hacen exactamente con la finalidad que se busca, ya que éstas permiten activarlas, pero no concretamente para avisar de una próxima inspección, sino que las activan para avisar de que ciertas piezas del vehículo deben ser sustituidas.

Por otro lado y como se puede observar, todas ellas muestran una baja complejidad de cara al usuario, lo cual es uno de los objetivos que también se buscará de cara al desarrollo de la aplicación.

Por todo ello, se considera que el desarrollo de la aplicación resulta algo novedoso, ya que, actualmente no existe ninguna aplicación que cumpla con las funcionalidades que se persiguen para el desarrollo de la misma.

En conclusión, tras la realización del análisis, se decide realizar una aplicación diferente a las vistas y en la que se cumpla con las siguientes funcionalidades:

- Cálculo de la fecha para la próxima inspección del vehículo.
- Activación de una notificación para avisar de la fecha de la inspección.
- Revisión pieza a pieza de las partes a revisar del vehículo.
- Información de la documentación necesaria para la inspección.

2.2 SISTEMA OPERATIVO SELECCIONADO

En este apartado se concretará con detalle cual es el sistema operativo que se ha elegido para el desarrollo de la aplicación y porque se ha elegido éste, dando una visión general de su funcionamiento, además de realizar un análisis del estado en el que se encuentran los demás sistemas operativos que existen en el mercado.

2.2.1 SISTEMAS OPERATIVOS EN EL MERCADO

Los sistemas operativos más comunes para dispositivos móviles que se pueden encontrar en el mercado son Android, Symbian OS, iOS, Blackberry OS, Windows Phone y Windows Mobile.

En la Figura 9 se puede observar el uso de los diferentes sistemas operativos y como Android muestra una clara ventaja sobre sus competencias llegando a alcanzar el 68,1% del total.

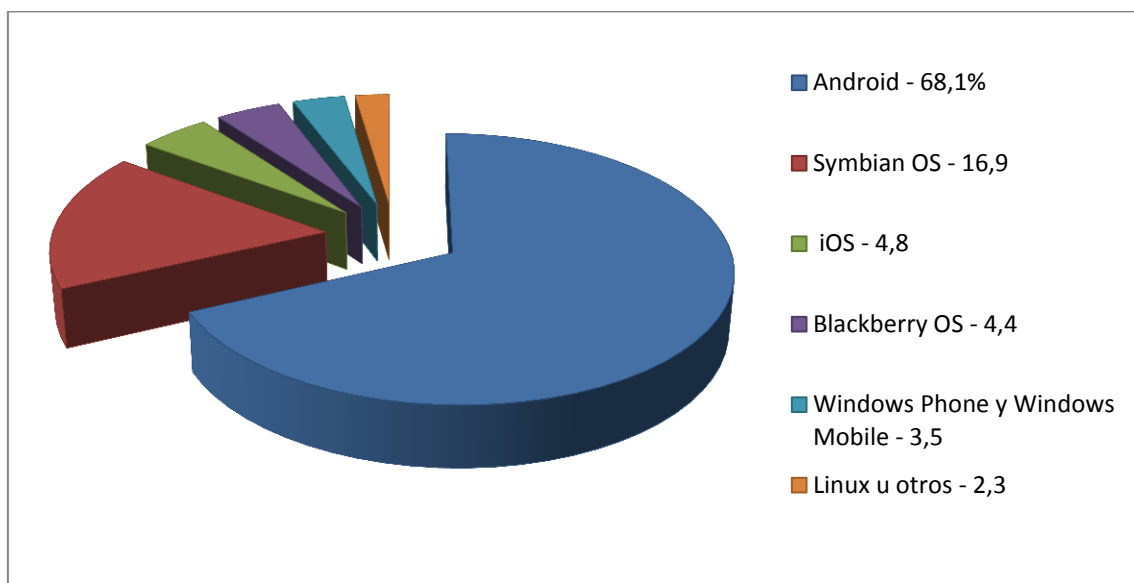


Figura 9: Porcentaje de uso de Sistemas operativos para móviles

Por tanto, al ver el porcentaje de dispositivos móviles con el sistema operativo de Android se decide escoger éste para llevar a cabo el desarrollo de la aplicación.

2.2.2 VERSIONES EN ANDROID

Android ha sufrido numerosas actualizaciones desde sus comienzos. Estas actualizaciones normalmente sirven para solucionar errores y añadir nuevas

funcionalidades. Por lo general, cada actualización se desarrolla bajo un nombre en código. En la Tabla 2 se pueden observar las diferentes versiones [8] junto con su nombre en código desde el comienzo en el mercado de éste sistema operativo.

Nombre	Código versión
Apple Pie	1.0
Banana Bread	1.1
Cupcake	1.5
Donut	1.6
Eclair	2.0 / 2.1
Froyo	2.2
Gingerbread	2.3 / 2.3.2 / 2.3.3 / 2.3.7
Honeycomb	3.1 / 3.2
Ice Cream Sandwich	4.0 / 4.0.2 / 4.0.3 / 4.0.4
Jelly Bean	4.1

Tabla 2: Versiones Android

La versión más actual del sistema operativo Android es la 4.1 “Jelly Bean” y ésta ha incorporado una gran cantidad funcionalidades. Como se puede observar en la Tabla 2, existe una gran cantidad de versiones para el sistema operativo Android y esto hace que se produzca una gran fragmentación debido a que todas estas versiones siguen siendo utilizadas por los usuarios.

Esta fragmentación se puede observar en la Figura 10, en la cual se puede ver el porcentaje de utilización de cada una de las diferentes versiones de Android.

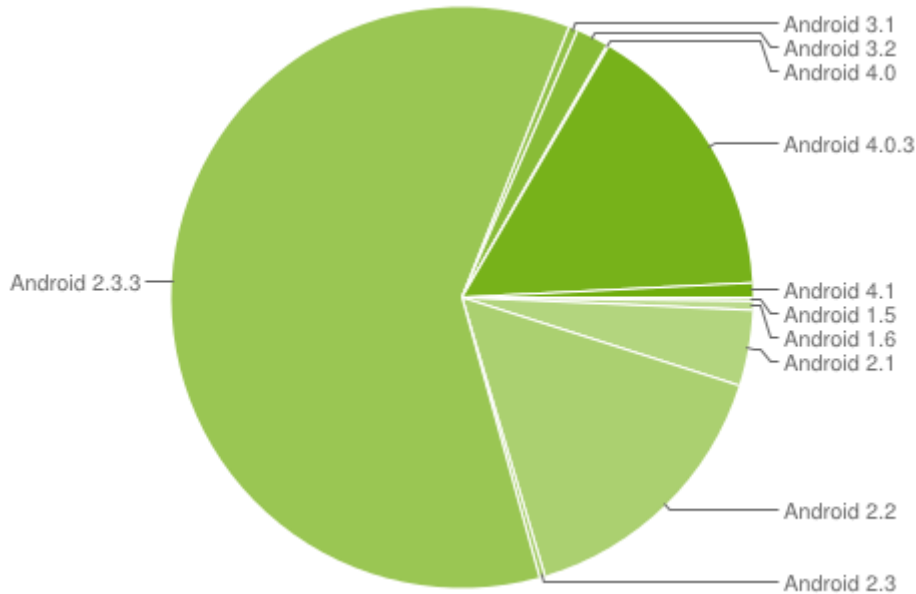


Figura 10: Uso diferentes versiones Android

Como se puede observar en la anterior imagen, la versión más utilizada actualmente es la versión 2.3.3 o también denominada “Gyngerbread”, la cual se encuentra actualmente en un 60,6% de los dispositivos basados en Android.

Por todo ello, se ha decidido que la versión que se utilizará para el desarrollo de la aplicación sea la 2.3.3, ya que se trata de la versión más extendida. Por ello la aplicación solo podrá ser utilizada en la versión 2.3.3 o superiores.

2.2.3 ARQUITECTURA DEL SISTEMA OPERATIVO ANDROID

Android es un sistema operativo móvil basado en Linux, que está enfocado para ser utilizado en dispositivos como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google.

Android está formado por varias capas que facilitan al desarrollador la creación de aplicaciones. Además, esta distribución permite acceder a las capas más bajas mediante el uso de diferentes librerías para que de esta forma no sea necesario que el desarrollador tenga que programar a bajo nivel.

La arquitectura de Android también es conocida como arquitectura en pila, ya que cada una de las capas utiliza elementos de la capa inferior para llevar a cabo sus funciones. Este tipo de arquitectura queda reflejada en la Figura 11.

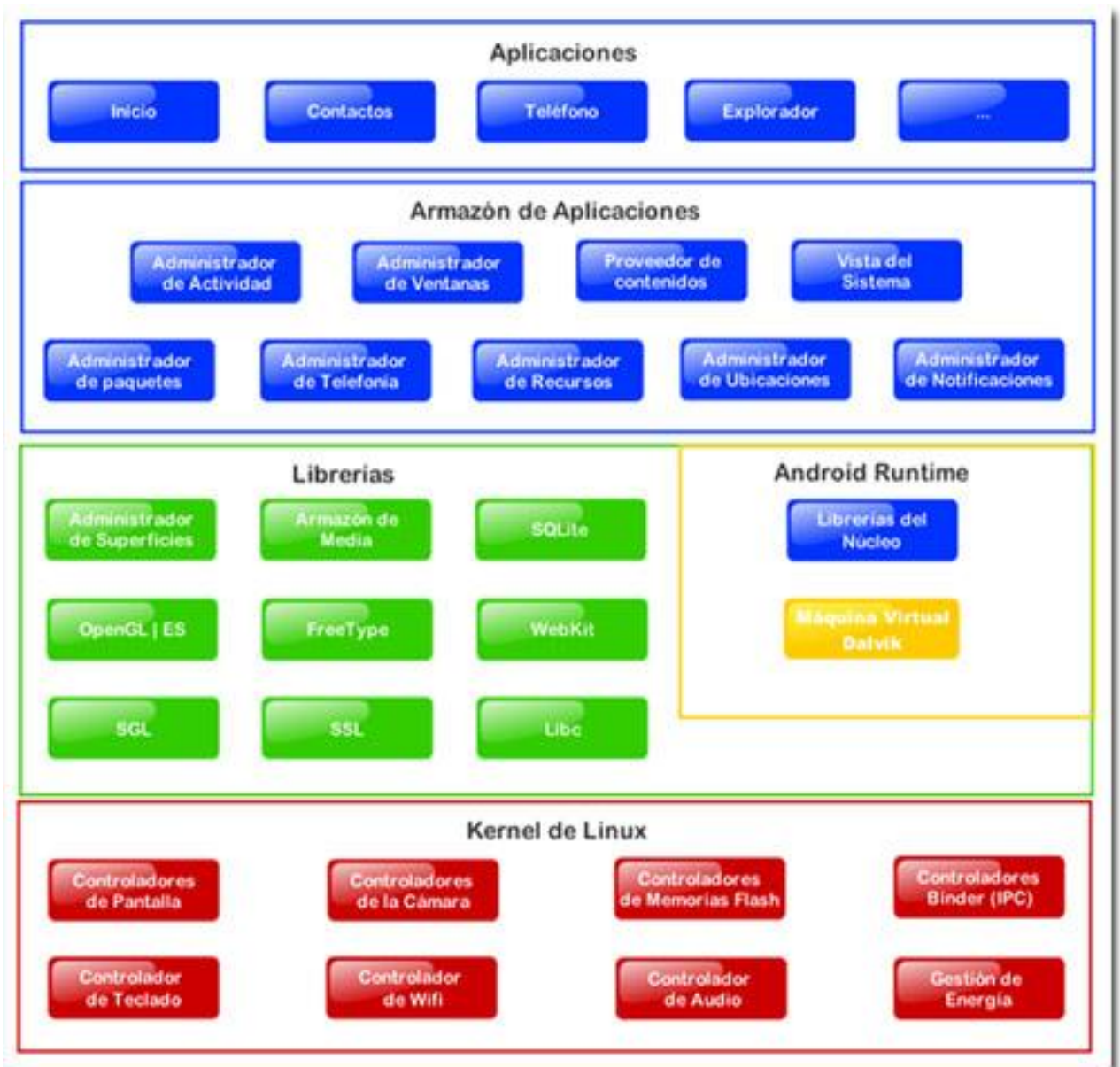


Figura 11: Arquitectura de Android

A continuación se detallará cada una de las capas vistas anteriormente iniciando de abajo hacia arriba.

2.2.3.1 Kernel de Linux

Tal y como se ha comentado anteriormente, el núcleo del sistema Android está basado en el kernel de Linux, similar al que se puede incluir en cualquier distribución de Linux, con la que única diferencia que éste es adaptado a las características del hardware en el que se quiere instalar, es decir, para dispositivos móviles.

El desarrollador no accede directamente a esta capa, sino que accederá directamente a las librerías que se encuentran en capas superiores.

El kernel se encargará de gestionar los diferentes recursos del teléfono, como la energía o la memoria, y del sistema operativo en sí, como los procesos o los elementos de comunicación.

2.2.3.2 Librerías

Esta capa está compuesta por las bibliotecas nativas de Android o también denominadas librerías. Todas ellas están escritas en C o C++ y compiladas para la arquitectura del hardware del teléfono en concreto.

El principal objetivo de las librerías es proporcionar la funcionalidad a las aplicaciones para tareas que se repiten continuamente, para evitar de esta forma tener que codificarlas cada vez que se ejecuten y garantizando que lo hacen de forma eficiente.

Entre las librerías más habituales se encuentra la de OpenGL (motor gráfico), Bibliotecas multimedia (para formatos de audio, imagen y video), Webkit (para el navegador), SSL (para el cifrado de comunicaciones), FreeType (para las fuentes de texto) o SQLite (para las bases de datos) entre otras.

2.2.3.3 Entorno de ejecución

Como se puede apreciar en la Figura 11 el entorno de ejecución no se considera una capa como tal, dado que también está formado por librerías.

El principal componente del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que dicha máquina virtual pueda ejecutarlas.

2.2.3.4 Framework de aplicaciones

Esta capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar su función. Los desarrolladores poseen un acceso completo a los mismos APIs del framework usados por las aplicaciones base. De este modo, la arquitectura de aplicación se diseña de forma que se pueda simplificar la reutilización de todos los componentes.

2.2.3.5 Aplicaciones

En esta última capa se incluyen todas las aplicaciones del dispositivo, tanto las que vienen preinstaladas en el dispositivo como aquellas que el usuario instala posteriormente.

2.2.4 CICLO DE VIDA DE LAS ACTIVIDADES EN ANDROID

En Android, el sistema operativo es el encargado de pausar, parar o destruir una aplicación según la necesidad que posea el dispositivo de recursos. Aun así es necesario que el desarrollador aprenda a utilizar dichos eventos para elaborar aplicaciones robustas y eficientes.

Para entender el comportamiento de las actividades en Android, es necesario comprender como funciona el ciclo de vida de las mismas, las cuales representan cada una de las pantallas de una aplicación. El ciclo de vida de una actividad [5] es el que se puede ver en la Figura 12.

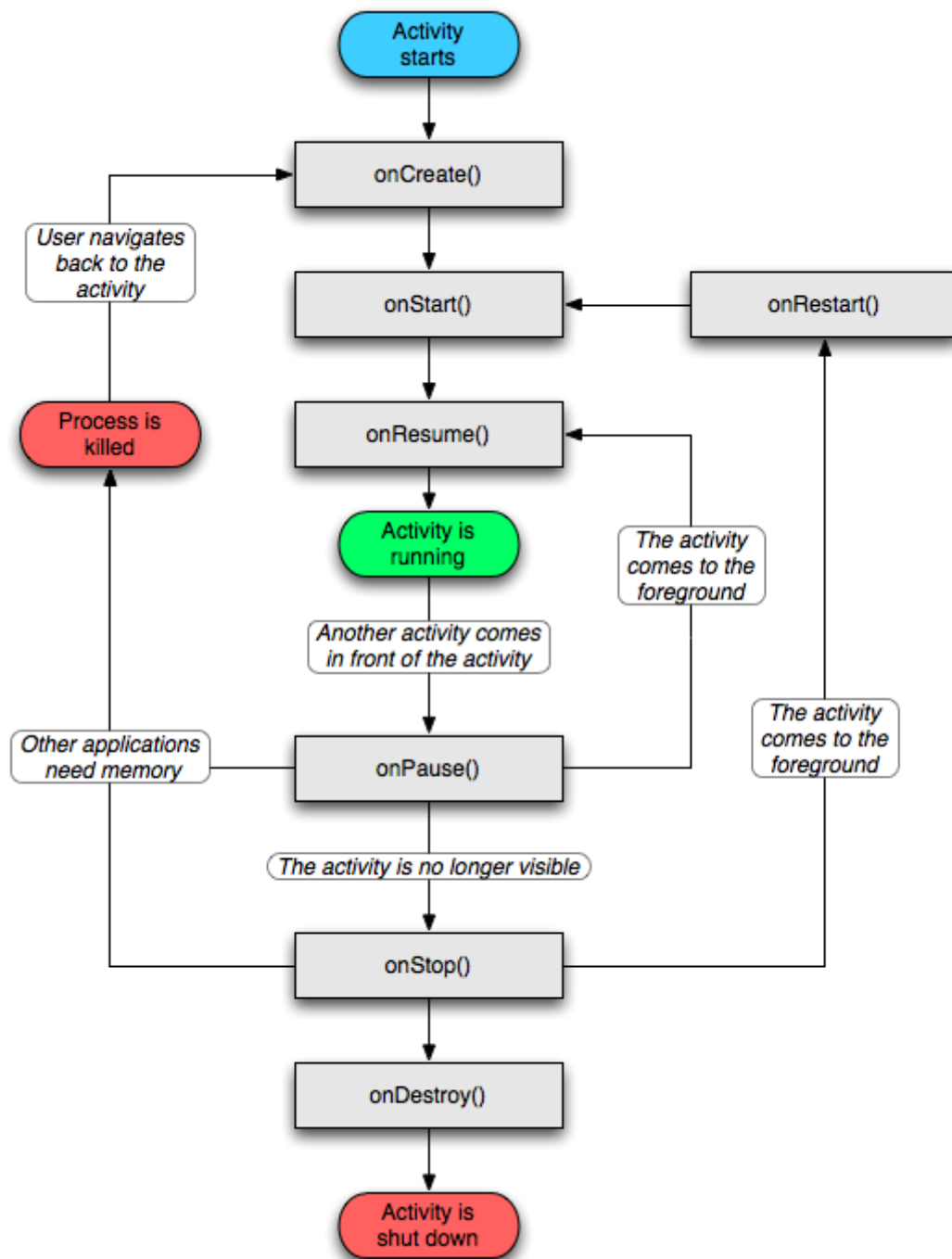


Figura 12: Ciclo de vida actividades en Android

A continuación se detalla el funcionamiento de cada uno de los métodos en el ciclo de vida de una actividad:

- **OnCreate():** Este método se inicia cuando la Actividad es llamada por primera vez. En este punto es donde se deberá crear la inicialización normal de la aplicación, generar las vistas o inicializar los datos, entre otras cosas. Después de esta llamada siempre se llama al método onStart().
- **OnRestart():** Este método es ejecutado cuando la Actividad ha sido parada, y se quiere volver a utilizar. Tal y como se puede observar en la Figura 12 después de un onStop() se ejecuta el método onRestart() e inmediatamente se llama al método onStart().
- **OnStart():** Se ejecuta cuando la Actividad se está mostrando en la pantalla del dispositivo.
- **OnResume():** Es ejecutado una vez que la Actividad se ha terminado de cargar en el dispositivo y el usuario comienza a interactuar con la aplicación. Una vez que el usuario deja de interactuar con dicha Actividad es cuando se llama al método onPause().
- **OnPause():** Este se ejecuta cuando el sistema inicia una nueva Actividad que necesitará recursos del sistema centrados en ella. Es necesario que la llamada a este método sea lo más rápida posible, ya que hasta que no se termine su ejecución el sistema no podrá iniciar la Actividad nueva. Después de esta llamada puede venir un onResume() si la Actividad que ha se llama con onPause() es la misma o un onStop() si se hace invisible para el usuario.
- **OnStop():** Este método es ejecutado cuando se quiere que una Actividad deje de ser visible para el usuario porque se quiere dar paso a otra nueva y dejarla en primer plano.
- **OnDestroy():** Esta es la llamada final de la Actividad, ya que una vez que se le llama, dicha Actividad es destruida. Si se quisiera arrancar de nuevo la Actividad será necesario comenzar desde el principio el ciclo de vida.

2.2.5 ENTORNO DE DESARROLLO INTEGRADO

Para llevar a cabo el desarrollo de la aplicación, el entorno de desarrollo integrado que se ha elegido ha sido *Eclipse Indigo* en su última versión, en el cual se ha instalado el SDK de Android [3] correspondiente.

Se eligió este entorno de desarrollo puesto que es el más común a la hora de desarrollar aplicaciones Android, y, además cuenta con un editor para interfaces gráficas que hace que su uso sea bastante intuitivo y fácil de manejar. Además, también cuenta con una máquina virtual que proporciona la opción de lanzar el

proyecto sobre un dispositivo virtual en caso de no disponer de uno físico para poder probar la aplicación.

Este entorno también cuenta con un asistente para la creación de archivos XML que hace que el desarrollo sea más fácil, ya que el desarrollador de esta forma no tiene que preocuparse tanto por la sintaxis.

Por último, Eclipse cuenta con un modo “Debug” que es muy interesante, ya que con él se podrá ver paso a paso las trazas que sigue el código gracias a marcas que se podrán poner donde más interese. Esto es de gran utilidad, ya que de esta manera se podrán solucionar numerosos problemas que en muchas ocasiones no se sabe de dónde proceden. A parte, Eclipse también cuenta con un complemento denominado *LogCat* que permitirá al desarrollador realizar trazas que también permitirá al desarrollador encontrar y solucionar errores de forma más rápida.

2.3 DESARROLLO ÁGIL COMO MÉTODO DE DESARROLLO DEL PROYECTO

El desarrollo ágil de software consiste en métodos de ingeniería del software que constituyen un nuevo enfoque en el desarrollo del mismo. Estos métodos son mejor aceptados por los desarrolladores de proyectos debido a la simplicidad de sus reglas y a su flexibilidad ante los cambios. Además dichos métodos están basados en el desarrollo iterativo e incremental.

Las metodologías ágiles de desarrollo se caracterizan por ser adaptativas en vez de predictivas, por lo cual, dichas metodologías no realizan planes en gran detalle durante largos periodos de tiempo, sino que se basan en los continuos cambios sobre el software, intentando por lo tanto ser procesos que se adaptan a todos estos cambios. La meta de estas metodologías por tanto es definir un proceso que funcionará de forma correcta con cualquiera que lo use.

2.3.1 METODOLOGIA ÁGIL SELECCIONADA

Existen numerosas metodologías para el desarrollo de software. La que se ha seleccionado para la realización del presente proyecto es la metodología Scrum [2].

Esta metodología se caracteriza por ser una metodología flexible y ágil que sirve para gestionar el desarrollo de software.

El principal objetivo de dicha metodología es devolver el máximo de la inversión para una empresa y se basa en construir primero la funcionalidad de mayor

valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación.

Con esta metodología el cliente se consigue el entusiasmo del cliente por el proyecto ya que ve como va creciendo paso a paso en cada iteración. Otro factor muy importante es que el cliente en cualquiera de las iteraciones puede realizar cambios funcionales o de prioridad.

Los principales beneficios que proporciona la metodología Scrum son los siguientes:

- **Cumplimiento de expectativas:** El cliente establece sus principales expectativas y de manera regular se comprueba junto con el equipo de desarrollo que todos ellos se cumplen.
- **Flexibilidad en los cambios:** Existe una alta capacidad de reacción ante los cambios que el cliente cree conveniente o por propias evoluciones del mercado.
- **Reducción del tiempo de uso:** El cliente tiene la posibilidad de comenzar a utilizar funcionalidades de la aplicación antes de que ésta esté finalizada.
- **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Conseguida gracias a la autonomía del equipo para organizarse.
- **Maximización del retorno de inversión:** Gracias a la producción de software que únicamente aportan mayor valor al negocio.
- **Predicciones de tiempo:** Es posible conocer con facilidad el tiempo que se tardará en disponer de una determinada funcionalidad.
- **Reducción de riesgos:** Gracias a la realización de las funcionalidades de más valor en primer valor y al conocimiento del tiempo que se tardará en realizar cada una de ellas, se permite despejar riesgos de una manera eficaz y anticipada.

En la siguiente imagen, Figura 13, se puede observar el transcurso que sigue la metodología Scrum.

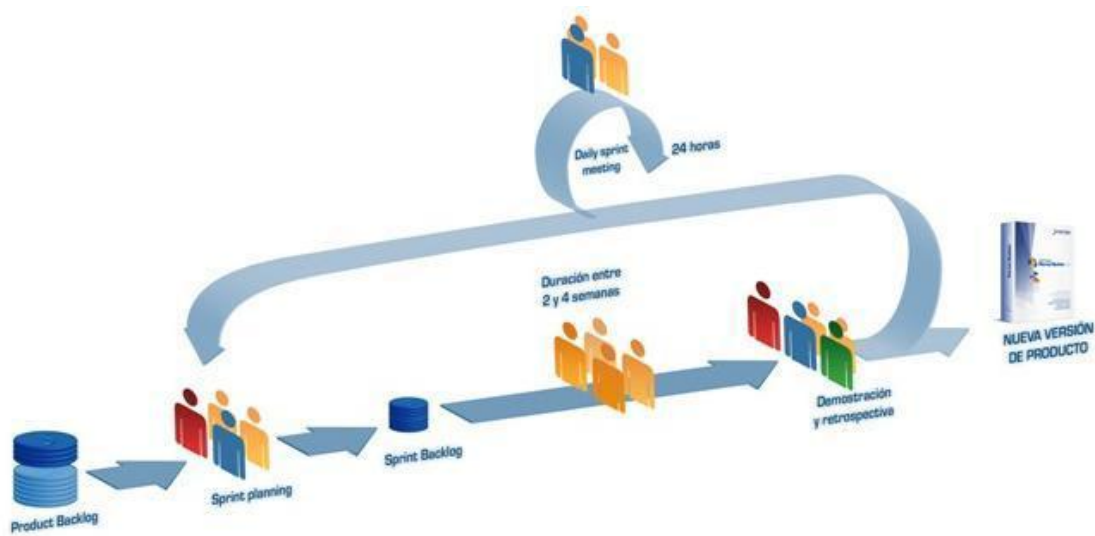


Figura 13: Diagrama metodología Scrum

2.4 CONCLUSIONES DEL ANÁLISIS

Tras el análisis llevado a cabo de otras aplicaciones similares, resulta interesante el propósito que se pretende conseguir de esta aplicación, ya que ofrece funcionalidades que no se han podido encontrar en aquellas otras.

Posiblemente, la funcionalidad que muestre más interés de cara a los usuarios potenciales, sea la posibilidad de añadir notificaciones para el recordatorio de las próximas inspecciones de los vehículos. Esto es así ya que, a pesar de que en la gran mayoría de los casos las personas suelen ser avisadas por medio de correo del deber de realizar dichas inspecciones, en algunas ocasiones dicho correo no llega, ya sea por el olvido, porque llega demasiado tarde o porque el usuario puede cambiar de domicilio y no quedar registrado como tal.

El hecho de no realizar la inspección técnica de un vehículo es de vital importancia, y no solo en cuanto al nivel de seguridad que el vehículo debe ofrecer a sus pasajeros, sino que también puede ser motivo de una multa económica e incluso de la propia inmovilización del vehículo si así lo creyeran conveniente las autoridades.

Por todo ello se piensa que esta funcionalidad es la que posee mayor importancia dentro de la aplicación.

Durante el periodo que se llevan a cabo estas tareas de análisis, se han llevado a cabo diferentes reuniones para comprobar el grado de avance del proyecto, e incluso

antes de la finalización de la propia tarea de análisis se ha comenzado también con la fase de diseño e implementación.

3 ESPECIFICACIÓN DE REQUISITOS

3	ESPECIFICACIÓN DE REQUISITOS	47
3.1	REQUISITOS DE USUARIO	48
3.1.1	<i>REQUISITOS FUNCIONALES.....</i>	<i>48</i>
3.1.2	<i>REQUISITOS DE RESTRICCIÓN</i>	<i>53</i>
3.2	CASOS DE USO	53

3.1 REQUISITOS DE USUARIO

En este apartado se van a desglosar los diferentes requisitos de usuario de la aplicación a desarrollar. Los diferentes requisitos se clasificarán en:

- Requisitos Funcionales.
- Requisitos de Restricciones de Diseño.

Todos estos requisitos se desarrollarán en tablas que tendrán el formato que se puede observar en la Tabla 3.

ID	Identificador
Nombre	Nombre de ejemplo
Dependencias	Otro Identificador
Descripción	Descripción breve y concisa del requisito.

Tabla 3: Formato tabla requisitos de usuario

El formato de los identificadores quedará definido por dos partes diferenciadas y separadas por un guion. Una primera en la que se mostrarán dos letras pertenecientes al tipo de requisito y una segunda en la que se le asignará un número iniciando para que sea único.

La parte izquierda del identificador quedará indicada por uno de los siguientes códigos:

- RF- Requisito Funcional.
- RR- Requisito de Restricción.

3.1.1 REQUISITOS FUNCIONALES

En este apartado se definirán los requisitos funcionales que serán los que sirvan para entender el funcionamiento de la aplicación y su comportamiento.

ID	RF - 001
Nombre	El usuario podrá revisar el vehículo
Dependencias	N/A
Descripción	Se debe dar la posibilidad de ver las partes del vehículo que son necesarias revisar.

Tabla 4: RF - 001 - Qué revisar en el vehículo

ID	RF - 002
Nombre	El usuario calculará la próxima revisión del vehículo
Dependencias	N/A
Descripción	Se ofrecerá la posibilidad de calcular la próxima inspección del vehículo, eligiendo para ello tanto el tipo de vehículo como la fecha de matriculación del mismo.

Tabla 5: RF - 002 - Próxima revisión del vehículo

ID	RF - 003
Nombre	El usuario verá las notificaciones guardadas
Dependencias	N/A
Descripción	El sistema ofrecerá la posibilidad de ver todas las notificaciones que hayan sido guardadas con anterioridad.

Tabla 6: RF - 003 - Ver Notificaciones

ID	RF - 004
Nombre	Ver documentación necesaria para la inspección
Dependencias	N/A
Descripción	El sistema ofrecerá la posibilidad de ver toda la documentación necesaria para la inspección del vehículo.

Tabla 7: RF - 004 - Ver documentación necesaria para la inspección

ID	RF – 005
Nombre	El usuario podrá revisar alumbrado del vehículo
Dependencias	RF – 001
Descripción	El sistema dará la opción de poder ver las partes del alumbrado del vehículo mediante una lista con las diferentes partes.

Tabla 8: RF - 005 - Revisar alumbrado

ID	RF – 006
Nombre	El usuario podrá revisar los neumáticos del vehículo
Dependencias	RF – 001
Descripción	El sistema dará la opción de realizar una revisión de los neumáticos del vehículo mediante una lista con sus diferentes partes.

Tabla 9: RF - 006 - Revisar neumático

ID	RF – 007
Nombre	El usuario podrá revisar los frenos del vehículo
Dependencias	RF – 001
Descripción	El sistema dará la opción de realizar una revisión de los frenos de su vehículo mediante una lista con las diferentes partes.

Tabla 10: RF - 007 - Revisar Frenos

ID	RF – 008
Nombre	El usuario podrá revisar generalmente su vehículo
Dependencias	RF – 001
Descripción	El sistema dará la opción de realizar una revisión general del vehículo mostrando una lista con las diferentes partes para ello.

Tabla 11: RF - 008 - Revisión general del vehículo

ID	RF – 009
Nombre	El usuario podrá seleccionar el tipo de vehículo
Dependencias	RF – 002
Descripción	El sistema dará la opción de elegir el tipo de vehículo para calcular la fecha de la próxima inspección.

Tabla 12: RF - 009 - Selección del tipo de vehículo

ID	RF – 010
Nombre	El usuario seleccionará la fecha de matriculación del vehículo
Dependencias	RF – 002
Descripción	El sistema dará la opción de seleccionar la fecha de matriculación del vehículo para realizar el cálculo de la próxima inspección.

Tabla 13: RF - 010 - Seleccionar fecha de matriculación vehículo

ID	RF – 011
Nombre	El usuario podrá ver los periodos de las inspecciones
Dependencias	RF – 002
Descripción	El sistema permitirá ver al usuario los periodos con los que cada vehículo debe realizar la inspección de su vehículo.

Tabla 14: RF - 011 - Mostrar periodos de las inspecciones

ID	RF – 012
Nombre	El usuario podrá añadir una notificación
Dependencias	RF – 002
Descripción	El sistema permitirá al usuario añadir una notificación, mediante su matrícula y la fecha seleccionada para ser avisado con antelación de la próxima inspección y serán mostradas en una tabla todas ellas.

Tabla 15: RF - 012 - Añadir notificación

ID	RF – 013
Nombre	El usuario podrá insertar la matrícula matricula del vehículo
Dependencias	RF – 002
Descripción	El sistema dará la opción al usuario de insertar la matrícula de su vehículo para que este quede identificado.

Tabla 16: RF - 013 – Insertar matricula vehículo

ID	RF – 014
Nombre	El usuario podrá elegir fecha notificación
Dependencias	RF – 002
Descripción	El sistema dará la opción al usuario de elegir la fecha de la notificación para ser avisado.

Tabla 17: RF - 014 - Elegir fecha notificación

ID	RF – 015
Nombre	El usuario eliminará las notificaciones
Dependencias	RF – 003, RF - 012
Descripción	El sistema permitirá al usuario eliminar cualquiera de las notificaciones que haya guardado con anterioridad.

Tabla 18: RF - 015 - Eliminar notificaciones

ID	RF – 016
Nombre	El usuario podrá retornar a la pantalla de inicio
Dependencias	N/A
Descripción	El sistema permitirá al usuario volver en cualquiera de las pantallas de la aplicación al menú inicial de la misma mediante un botón.

Tabla 19: RF - 017 - Retornar a la pantalla de inicio

3.1.2 REQUISITOS DE RESTRICCIÓN

En este apartado se definirán los requisitos de restricción de la aplicación. Estos requisitos marcarán las limitaciones que poseerá la aplicación. El formato de tabla seguido para estos requisitos es el mismo que en los casos anteriores.

ID	RR – 001
Nombre	Versión del Sistema Operativo
Prioridad	Normal
Descripción	La aplicación deberá ser compatible con todos los dispositivos Android basados en versiones 2.3 “Gingerbread” o posteriores.

Tabla 20: RR - 001 - Versión del Sistema Operativo

ID	RR – 002
Nombre	Versión del SDK de Java
Dependencias	N/A
Descripción	El lenguaje de programación que se utilizará para la aplicación será JAVA y el nivel de compilación del mismo será el 1.6.

Tabla 21: RR - 002 - Versión del SDK de Java

3.2 CASOS DE USO

En este apartado se muestran los casos de uso, los cuales van a servir para identificar la relación existente entre el actor (usuario que utilice la aplicación) y el software (la aplicación). Dichos casos de uso son los que se muestran en la Figura 14.

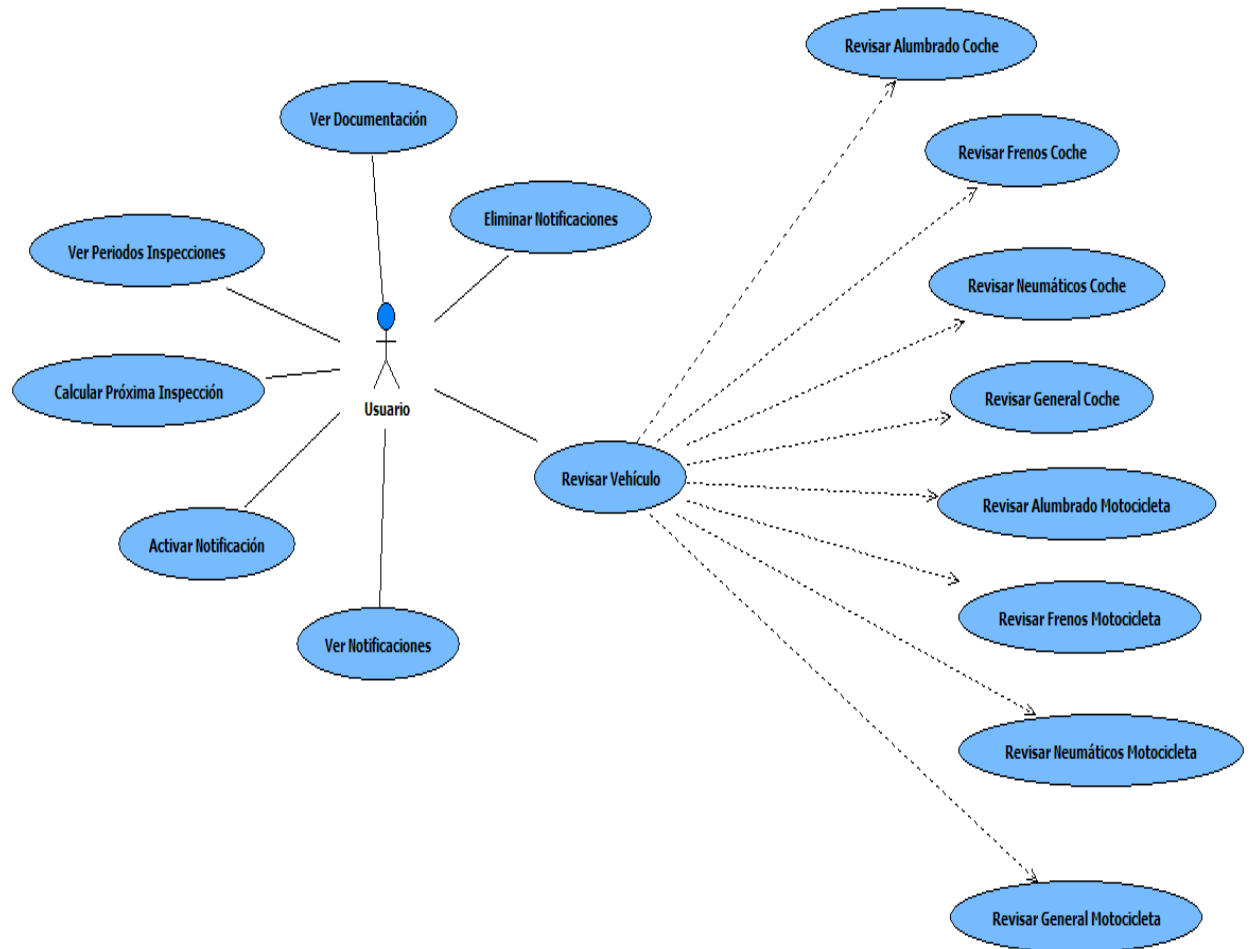


Figura 14: Casos de Uso

A continuación se representarán los casos de uso mediante tablas definiendo sus atributos:

- **ID:** Muestra el caso de uso (CU) seguido de un valor numérico que será único.
- **Nombre:** Título breve que identificará el nombre del caso de uso.
- **Actor:** Actor principal del caso de uso.
- **Objetivo:** Finalidad pretendida por el caso de uso.
- **Pasos:** Acciones que se deben llevar a cabo para realizar el caso de uso.
- **Precondiciones:** Condiciones previas que se deben cumplir para que ocurra el caso de uso.

- **Postcondiciones:** Condiciones producidas después de que se produzca el caso de uso.
- **Requisitos de usuario relacionados:** Requisitos que están relacionados con el caso de uso concreto.

ID	CU-01	NOMBRE	REVISAR VEHICULO
Actor		Usuario.	
Objetivo		Revisar las diferentes partes de un vehículo.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar".	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación.	
Postcondiciones		La aplicación mostrará el tipo de vehículo que se desea revisar.	
Requisitos de usuario relacionados		RF - 001	

Tabla 22: CU - 01 - Revisar partes vehículo

ID	CU-02	NOMBRE	REVISAR ALUMBRADO COCHE
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del alumbrado.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar el coche como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar el alumbrado.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de coche y haber elegido a la actividad de revisar alumbrado.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de alumbrado dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 005	

Tabla 23: CU-02 – Revisar las diferentes partes del alumbrado coche

ID	CU-03	NOMBRE	REVISAR FRENOS COCHE
Actor		Usuario.	
Objetivo		Revisar las diferentes partes de los frenos.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar el coche como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar los frenos.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de coche y haber elegido a la actividad de revisar frenos.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de frenos dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF -001, RF – RF – 006	

Tabla 24: CU-03 - Revisar las diferentes partes de los frenos coche

ID	CU-04	NOMBRE	REVISAR NEUMÁTICOS COCHE
Actor		Usuario.	
Objetivo		Revisar las diferentes partes de los neumáticos.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar el coche como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar neumáticos.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de coche y haber elegido a la actividad de revisar neumáticos.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de neumáticos dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 007	

Tabla 25: CU-04 - Revisar las diferentes partes de los neumáticos coche

ID	CU-05	NOMBRE	REVISAR GENERAL COCHE
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del vehículo en general.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar el coche como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar general.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de coche y haber elegido a la actividad de revisar general	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de general dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 008	

Tabla 26: CU-05 - Revisar las diferentes partes coche en general

ID	CU-06	NOMBRE	REVISAR ALUMBRADO MOTICICLETA
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del vehículo en general.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción “Qué revisar”. 3- Seleccionar la motocicleta como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar el alumbrado.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de motocicleta y haber elegido a la actividad de revisar alumbrado.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de general dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 005	

Tabla 27: CU – 06 - Revisar partes alumbrado motocicleta

ID	CU-07	NOMBRE	REVISAR FRENOS MOTOCICLETA
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del vehículo en general.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar la motocicleta como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar frenos.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de motocicleta y haber elegido a la actividad de revisar frenos.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de general dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF -001, RF – RF – 006	

Tabla 28: CU- 07 - Revisar partes freno motocicleta

ID	CU-08	NOMBRE	REVISAR NEUMATICOS MOTOCICLETA
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del vehículo en general.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar la motocicleta como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar neumáticos.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de motocicleta y haber elegido a la actividad de revisar neumáticos.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de general dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 007	

Tabla 29: CU- 08 - Revisar partes neumáticos motocicleta

ID	CU-09	NOMBRE	REVISAR GENERAL MOTOCICLETA
Actor		Usuario.	
Objetivo		Revisar las diferentes partes del vehículo en general.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Qué revisar". 3- Seleccionar la motocicleta como tipo de vehículo a revisar. 4- Seleccionar la opción de revisar general.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, haber seleccionado la opción de motocicleta y haber elegido a la actividad de revisar general.	
Postcondiciones		Se actualizará el valor de cada elemento de la lista de general dependiendo del estado de cada uno de ellos.	
Requisitos de usuario relacionados		RF – 001, RF – 008	

Tabla 30: CU- 09 - Revisar partes general motocicleta

ID	CU-10	NOMBRE	CALCULAR PROXIMA INSPECCION
Actor		Usuario.	
Objetivo		Calcula la fecha de la próxima inspección del vehículo deseado a partir de los datos introducidos.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción “Próxima revisión”. 3- Introducir los datos solicitados del vehículo. 4- Pulsar sobre el botón calcular.	
Precondiciones		El usuario debe haber iniciado correctamente la aplicación, acceder a la actividad de calcular la próxima inspección e introducir los datos correctamente.	
Postcondiciones		Se obtendrá la fecha de la próxima inspección para el vehículo.	
Requisitos de usuario relacionados		RF – 002, RF – 009, RF – 010	

Tabla 31: Cu-10 - Calcula de fecha próxima inspección

ID	CU-11	NOMBRE	ACTIVAR NOTIFICACION
Actor		Usuario.	
Objetivo		Activar notificación.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción “Próxima revisión”. 3- Introducir los datos del vehículo solicitados. 4- Pulsar sobre el botón calcular. 5- Pulsar sobre la opción de activar notificación.	
Precondiciones		El usuario deberá haber iniciado correctamente la aplicación y haber calculado correctamente la próxima inspección de su vehículo	
Postcondiciones		Se guardará una notificación para ser avisado de la próxima inspección.	
Requisitos de usuario relacionados		RF – 012, RF – 013, RF – 014	

Tabla 32: CU-11 - Activar notificación

ID	CU-12	NOMBRE	VER PERIODOS INSPECCIONES
Actor		Usuario.	
Objetivo		Ver los periodos con los que cada vehículo debe realizar la inspección de su vehículo.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Próxima revisión". 3- Pulsar sobre el botón de información.	
Precondiciones		El usuario deberá haber iniciado correctamente la aplicación, accederá a la actividad de próxima revisión.	
Postcondiciones		Se le proporcionará toda la información sobre los periodos de inspección de cada vehículo.	
Requisitos de usuario relacionados		RF – 011	

Tabla 33: CU-12 - Ver periodos inspección vehículos

ID	CU-13	NOMBRE	VER NOTIFICACIONES
Actor		Usuario.	
Objetivo		Ver las notificaciones que han sido previamente almacenadas.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Mis Notificaciones".	
Precondiciones		El usuario deberá haber iniciado correctamente la aplicación, y deberá acceder a la actividad para ver las notificaciones.	
Postcondiciones		Serán mostradas todas las notificaciones que han sido guardadas y que no han sido eliminadas.	
Requisitos de usuario relacionados		RF – 012, RF – 003	

Tabla 34: CU-13 - Ver notificaciones almacenadas

ID	CU-14	NOMBRE	ELIMINAR NOTIFICACIONES
Actor		Usuario.	
Objetivo		Eliminar notificaciones que se desee.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Próxima revisión". 3- Pulsar sobre el botón eliminar notificación. 4- Confirmar la eliminación de la notificación.	
Precondiciones		El usuario deberá haber iniciado correctamente la aplicación y deberá haber guardado previamente alguna notificación.	
Postcondiciones		Se eliminará la notificación que el usuario crea conveniente.	
Requisitos de usuario relacionados		RF – 003, RF 012, RF – 015	

Tabla 35: CU-14 - Eliminar notificaciones

ID	CU-15	NOMBRE	VER DOCUMENTACION
Actor		Usuario.	
Objetivo		Ver la documentación necesaria para poder pasar la inspección satisfactoriamente.	
Pasos		1- Lanzar la aplicación. 2- Elegir la opción "Documentación Necesaria".	
Precondiciones		El usuario deberá haber iniciado la aplicación correctamente y haber accedido a la actividad correspondiente con la documentación del vehículo.	
Postcondiciones		Se mostrará al usuario la documentación necesaria para poder pasar la inspección de su vehículo.	
Requisitos de usuario relacionados		RF - 004	

Tabla 36: CU-15 - Ver documentación necesaria

4 DISEÑO

4.1	DISEÑO EXTERNO.....	68
4.1.1	ALTERNATIVAS DE DISEÑO INTERNO.....	68
4.1.1.1	DISEÑO INICIAL.....	68
4.1.1.2	DISEÑO EXTERNO FINAL.....	75
4.2	DISEÑO INTERNO	91
4.2.1	ALTERNATIVAS DISEÑO INTERNO	91
4.2.1.1	DISEÑO SIN BASE DE DATOS.....	91
4.2.1.2	DISEÑO CON BASE DE DATOS REMOTA	91
4.2.1.3	DISEÑO CON BASE DE DATOS NO REMOTA	92
4.2.2	DIAGRAMA DE COMPONENTES	92
4.2.3	DIAGRAMA DE SECUENCIA	93
4.2.4	DIAGRAMA DE CLASES.....	104
4.2.5	MODELO ENTIDAD-RELACIÓN BASE DE DATOS	127
4.2.6	DISEÑO DEL MODELO FÍSICO DE DATOS.....	128

Este capítulo servirá para tener una visión del diseño de la aplicación. En él se diferenciarán dos apartados, por un lado el diseño externo, y por otro el diseño interno de la aplicación.

4.1 DISEÑO EXTERNO

El diseño externo de la aplicación servirá para dar una visión de como va a ser la aplicación externamente, es decir, como va a ser su apariencia de cara al usuario. Se mostrarán las alternativas de diseño obtenidas, en las que aparecerá el diseño externo inicial y el diseño externo final.

4.1.1 ALTERNATIVAS DE DISEÑO INTERNO

A continuación se procederá a ver las diferentes alternativas de diseño sobre las que se ha trabajado. Existen dos alternativas de diseño interno, una sobre la que se trabajo inicialmente, y otra final que fue la evolución de la primera.

4.1.1.1 DISEÑO INICIAL

A continuación se observará el Paper Prototype inicial sobre el que se trabajó.

Como se puede observar en Figura 15 - Pantalla inicial primer prototipo, inicialmente existían seis funcionalidades. Dos de estas funcionalidades fueron suprimidas nada más iniciar el proyecto, a parte de por el escaso tiempo del que se disponía para el desarrollo de la aplicación, por que el cliente decidió que no eran imprescindibles y que se podrían omitir.



Figura 15 - Pantalla inicial primer prototipo

La siguiente pantalla Figura 16 - Pantalla que revisar primer prototipo es a la que se accede pulsando la opción “¿Qué revisar?”. Cuenta con el botón de inicio rápido para volver a la pantalla inicial de manera instantánea.

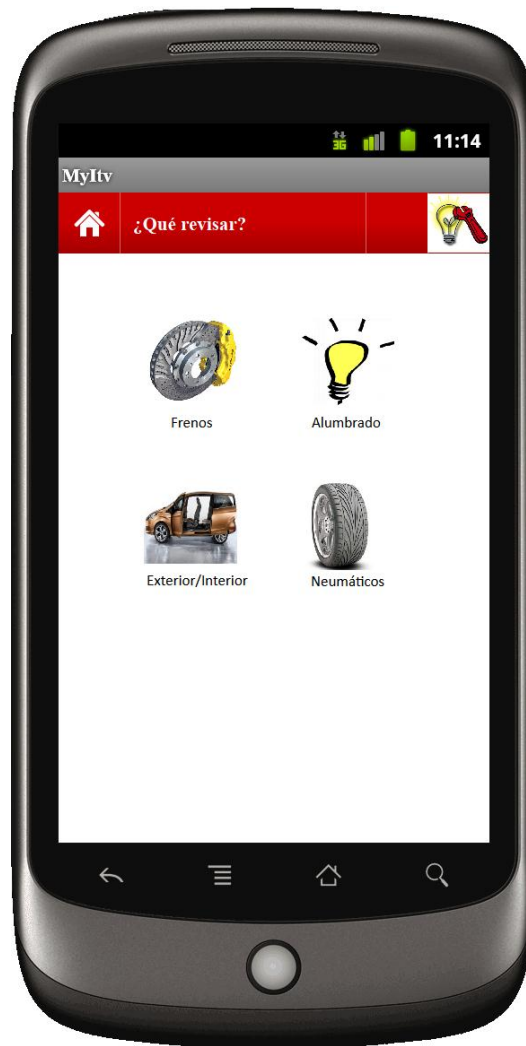


Figura 16 - Pantalla que revisar primer prototipo

La siguiente pantalla Figura 17 - Pantalla revisar frenos primer prototipo es a la que se accede tras pulsar alguna de las opciones de la pantalla anteriormente vista. Se accede, por ejemplo, a la opción “Frenos”.

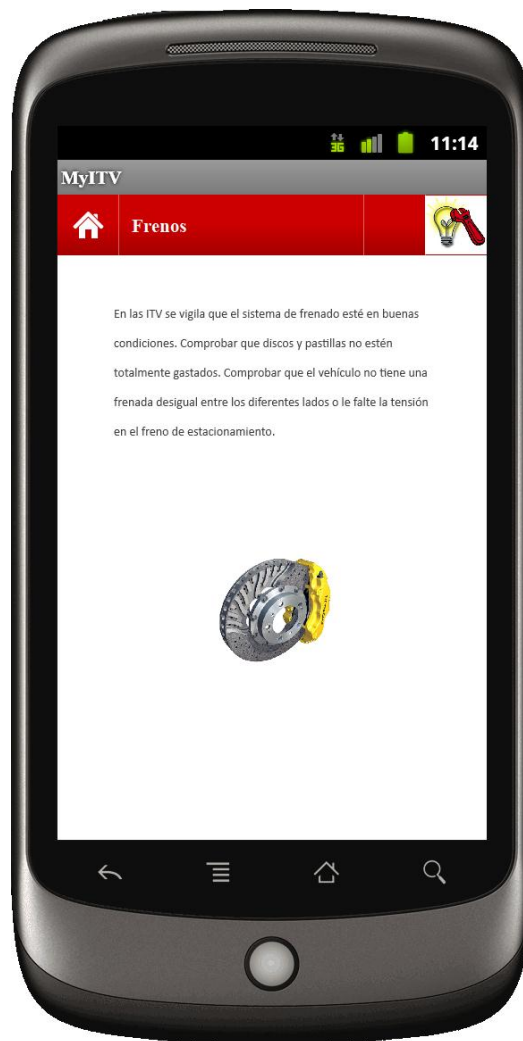


Figura 17 - Pantalla revisar frenos primer prototipo

La siguiente opción que se podría elegir desde la pantalla inicial sería la de la documentación necesaria, la cual se puede ver en la Figura 18. En esta pantalla se muestra al usuario toda la documentación que necesitará para poder afrontar la inspección de su vehículo con éxito.

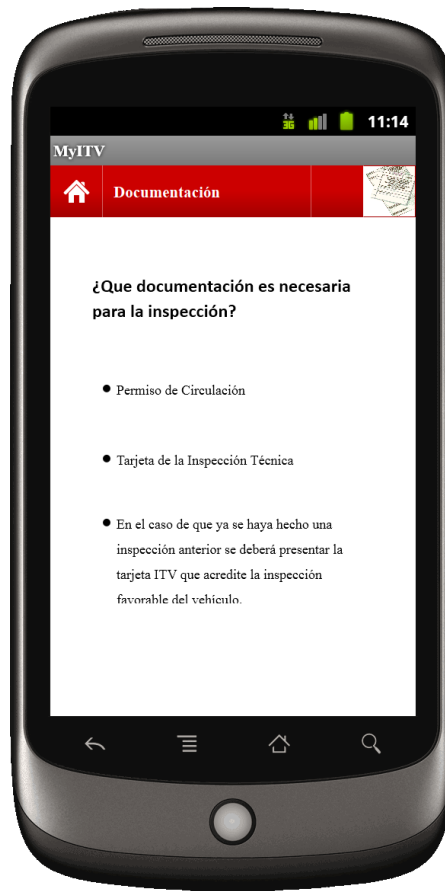


Figura 18 - Pantalla documentación necesaria primer prototipo

La siguiente opción que se podría elegir desde la pantalla principal sería la de “Próxima revisión”, Figura 19. Con esta opción el usuario podrá calcular la fecha de la siguiente revisión para su vehículo.



Figura 19 - Pantalla próxima revisión primer prototipo

Tras introducir los datos necesarios y pulsar sobre el botón calcular se accederá a la siguiente pantalla, Figura 20. En esta pantalla se introducirán los datos necesarios para activar la notificación que avisará de la inspección pendiente.



Figura 20 - Pantalla activar recordatorio primer prototipo

La siguiente opción que se podría elegir desde la pantalla inicial sería la de “Recordatorios”, Figura 21. En ella se mostrarán todas las notificaciones guardadas.

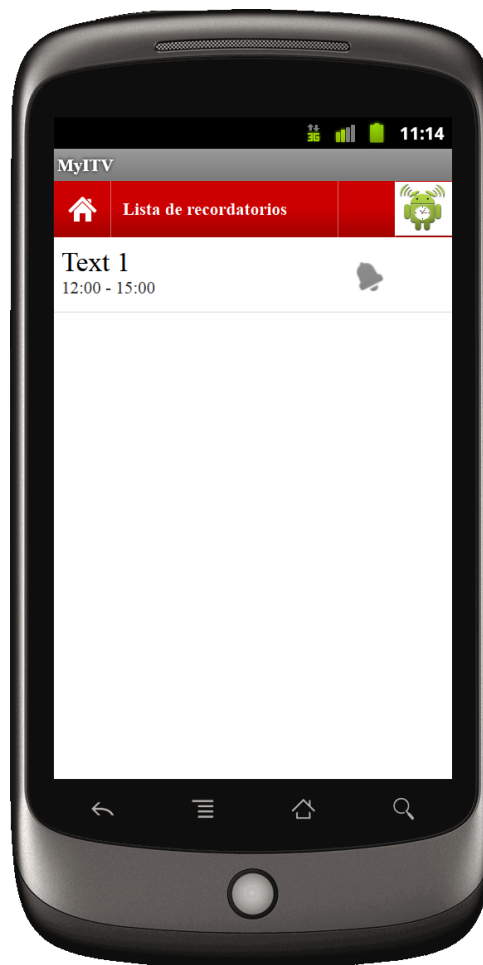


Figura 21 - Pantalla recordatorios primer prototipo

Las demás opciones restantes que se pueden elegir desde la pantalla principal no fueron desarrolladas finalmente, por lo que no se considera necesario explicarlas en el presente apartado.

4.1.1.2 DISEÑO EXTERNO FINAL

En este apartado se mostrará el *paper prototype* final de la aplicación, que tal y como se comentó anteriormente, servirá para expresar una idea del diseño lo más rápidamente posible y que además servirá para ayudar con la generación y evaluación de las ideas de diseño.

A continuación se mostrará el prototipo diseñado, el cual guardará bastante relación con la aplicación final, pero que en algunos casos puede llegar a diferir en pequeños detalles.

En la Figura 22 se muestra la que será la actividad principal de la aplicación. En ella se pueden observar las cuatro funcionalidades principales de las que va a constar la aplicación. Cada imagen consta de un botón que al ser pulsado llevará a una nueva actividad.

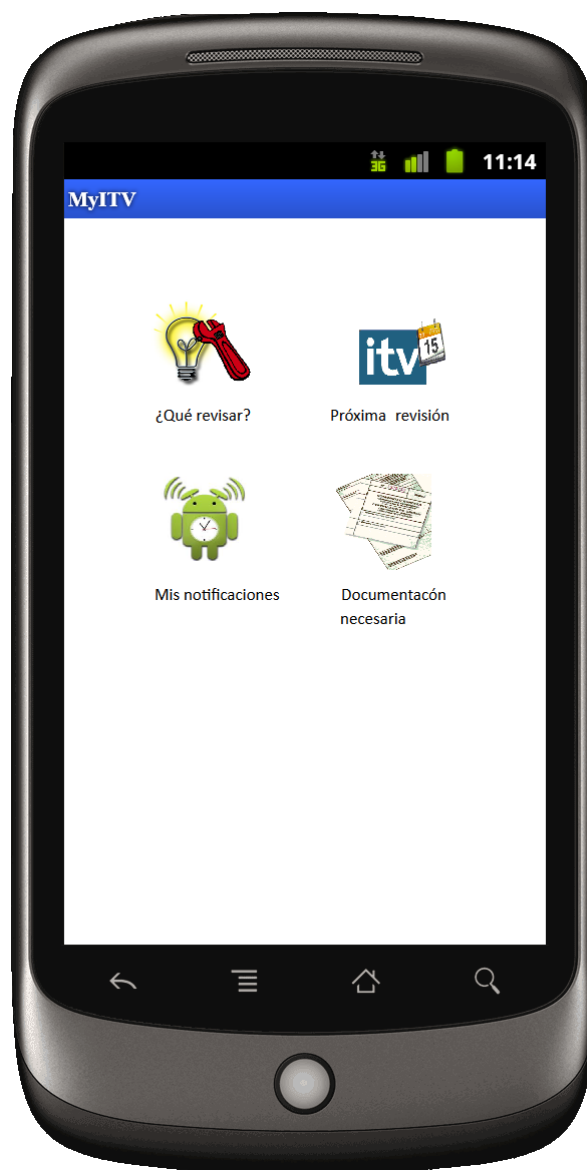


Figura 22: Pantalla inicial

En primer lugar se pulsará sobre el botón “¿Qué revisar?”, el cual nos llevará a la actividad que se puede observar en la Figura 23. Una vez aquí se puede observar como aparecen otras cuatro nuevas opciones para elegir. Además, en la barra de acciones se puede observar como aparece el icono de una casa. Este icono aparecerá en la mayoría de las actividades y servirá para volver a la pantalla inicial en el momento que se desee. Además también aparece en esta nueva barra de acción el nombre de la actividad en la que nos encontramos.

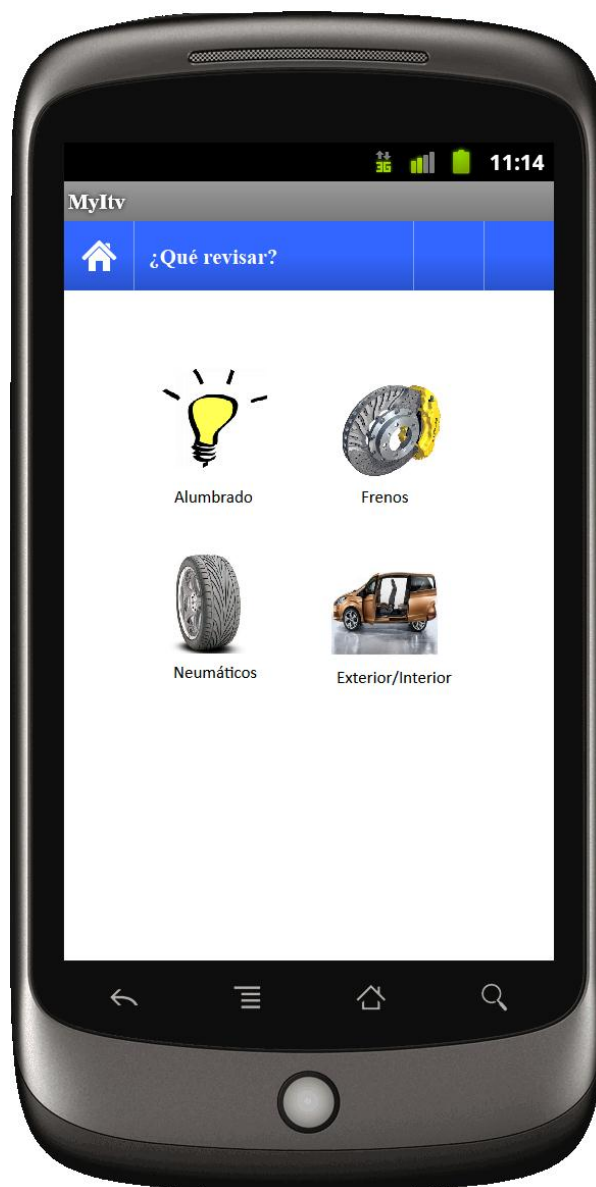


Figura 23: Pantalla Que revisar

Una vez en esta nueva actividad, pulsaremos sobre el primer botón de “Alumbrado”. Dicho botón nos llevará a una nueva actividad que es la que se puede observar en la Figura 24. En esta nueva actividad se puede ver que aparece una lista con todas las partes del alumbrado que deberemos revisar del vehículo. Todas ellas inicialmente aparecen marcadas con un símbolo negativo.



Figura 24: Pantalla lista Alumbrado

Para poder cambiar estos símbolos negativos deberemos acceder a cada una de las partes del vehículo. Tras acceder al primer elemento de la lista “Posición” accederemos a una nueva actividad, que es la que se puede observar en la Figura 25.



Figura 25: Pantalla primera alumbrado

En esta nueva actividad, la aplicación mandará un mensaje, y para acceder a la siguiente actividad habrá que pulsar sobre la flecha azul. Una vez pulsada dicha flecha, se accederá a la nueva actividad que es la que se puede observar en la Figura 26.



Figura 26: Pantalla segunda alumbrado

Una vez en esta nueva actividad nos aparecerán dos opciones. Una para marcar la respuesta como verdadera y otra como falsa. Al marcar la respuesta como verdadera la lista que se cargó anteriormente quedará modificada y aparecerá de la forma que muestra la Figura 27.

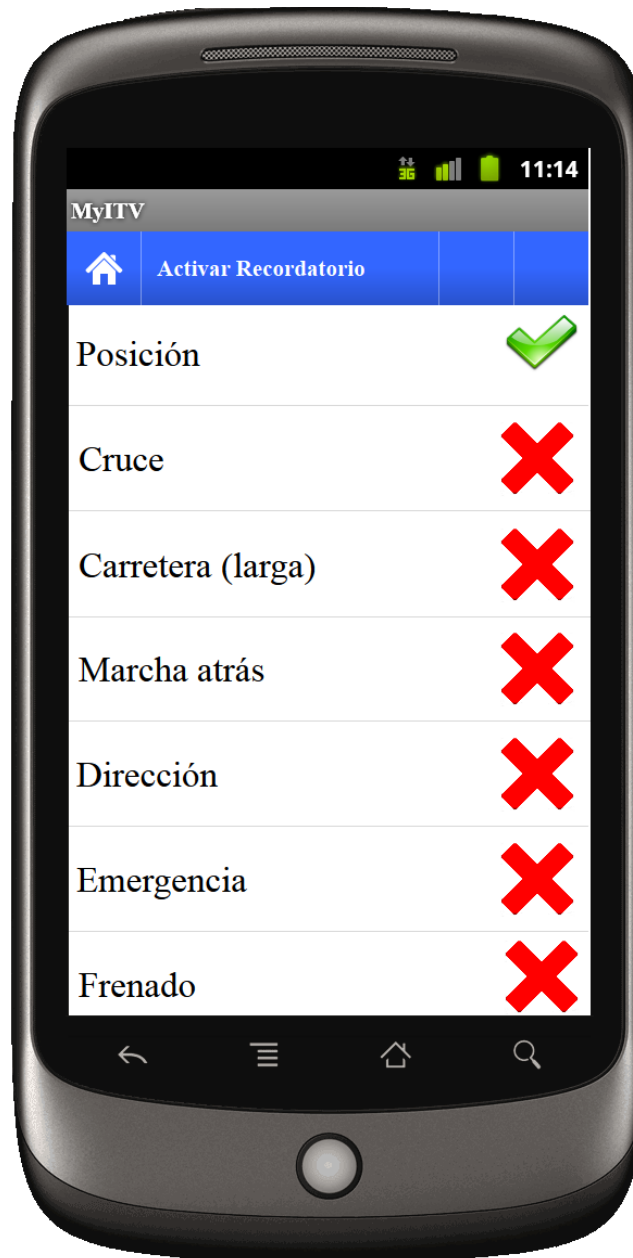


Figura 27: Pantalla lista modificada alumbrado

Este mismo paso se podrá repetir con todos los demás elementos de la lista.

Al igual que con el “Alumbrado” del vehículo, estos pasos serán los mismos para las otras partes del vehículo que se mostraban en la Figura 23.

Una vez revisadas las partes del vehículo se volverá a la actividad principal, Figura 22, y ahora se pulsará sobre el botón “Próxima revisión”. Al pulsar dicho botón se accederá sobre una nueva actividad que será la de la Figura 28.



Figura 28: Pantalla próxima revisión

En esta nueva actividad aparecen nuevas opciones. Entre ellas aparece una lista desplegable en la que se podrá seleccionar el tipo de vehículo sobre el que queramos calcular la fecha de revisión, un calendario en el que introducir la propia fecha de matriculación, un botón para calcular la fecha y un botón de información en la barra de acción que servirá para mostrar los periodos de revisión que tienen cada uno de los tipos de vehículos.

Se pulsará el botón de información para pasar a la siguiente actividad que es la que será la que se muestra en la Figura 29.



Figura 29: Pantalla información periodos inspección

Una vez vistos los periodos de inspección de cada vehículo se procederá a calcular la fecha para un vehículo. Para ello se volverá a la actividad anterior, Figura 28 y se introducirá tanto el tipo de vehículo como la fecha de matriculación del mismo. Una vez introducidos dichos datos se pulsará el botón de calcular. Al pulsar este botón se accederá a la siguiente actividad que es la que se muestra en la Figura 30.

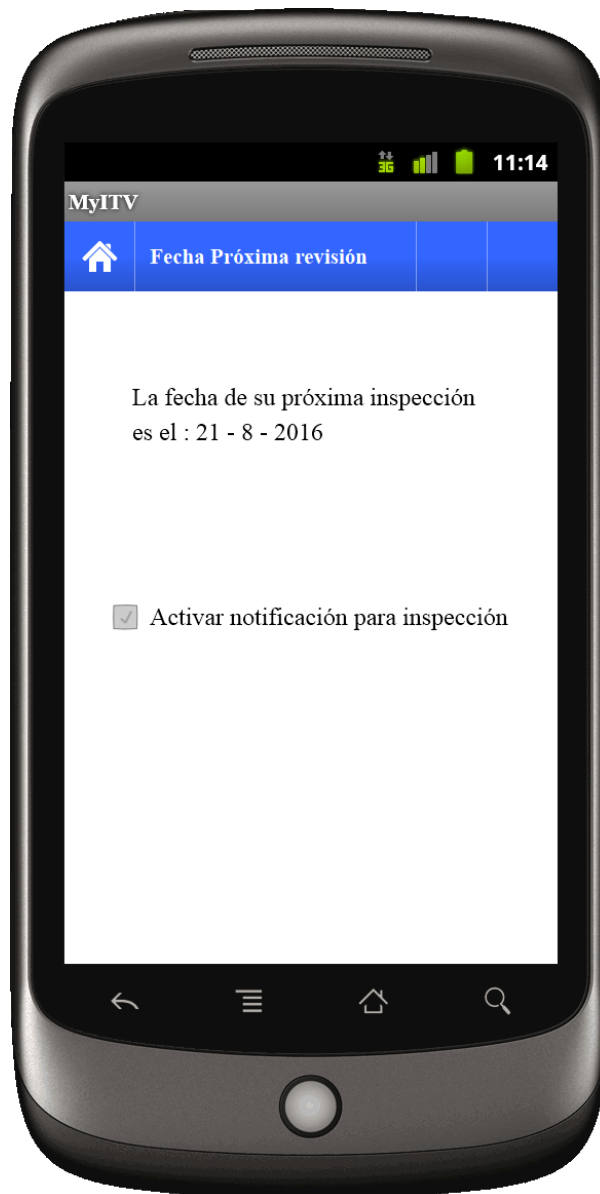


Figura 30: Pantalla activar notificación

En esta nueva actividad aparecerá la fecha en la que se tiene que pasar la próxima revisión. Además, también aparecerá una opción que servirá para proceder a activar una notificación para alertar al usuario días antes de la revisión de su vehículo. Al pulsar en la opción de activar la notificación se accederá a la actividad que se muestra en la Figura 31.



Figura 31: Pantalla diálogo confirmar notificación

En esta nueva actividad aparecerá una alerta en la que se deberá introducir tanto la matrícula del vehículo, como la fecha en la que se desea ser avisado. En la fecha de recordatorio aparecerá una lista desplegable en la que se mostrarán varias opciones para elegir con cuantos días de antelación desea ser avisado el usuario.

Una vez introducidos los datos, existirán otros dos botones, uno para aceptar mediante el cual quedará almacenada la notificación, y otro para cancelar, mediante el cual se volverá de nuevo a la actividad anterior, Figura 28.

Tal y como se ha comentado anteriormente, una vez que se pulsa el botón aceptar, la notificación quedará almacenada, y para poder comprobarlo se podrá pulsar sobre el botón “Mi Calendario” que aparece en la pantalla inicial de la aplicación, Figura 22.

Una vez pulsado dicho botón se accederá a la actividad que se muestra en la Figura 32.

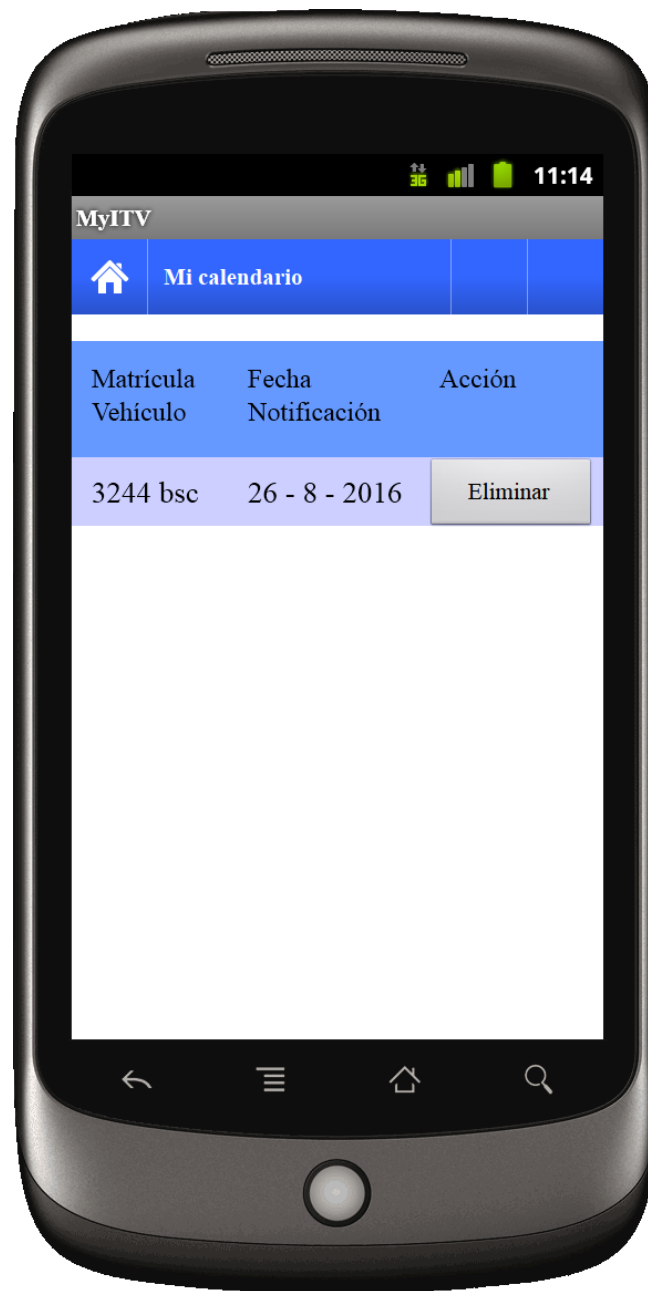


Figura 32: Pantalla ver notificaciones

Como se puede observar, en esta actividad se muestra una tabla con todas las notificaciones existentes. Además, en esta actividad también existe un botón que servirá para eliminar las notificaciones que se desee una vez que sea pulsado.

Para concluir, la última opción que se podría elegir en la actividad principal, Figura 22, es la de “Documentación necesaria”. Una vez pulsado dicho botón accederemos a la actividad que muestra la Figura 33.



Figura 33: Pantalla ver documentación 1

Desde esta actividad se podrá acceder a la siguiente mediante la flecha que se encuentra en la parte inferior derecha de la pantalla.

Una vez en la nueva actividad Figura 34, se mostrará la siguiente documentación necesaria. Se podrá volver a la anterior actividad mediante la flecha que se encuentra en la parte inferior izquierda de la aplicación o acceder a la siguiente mediante la que se encuentra en la parte inferior derecha.



Figura 34: Pantalla ver documentación 2

Si se accede a la siguiente actividad se mostrará la última documentación necesaria Figura 35.



Figura 35: Pantalla ver documentación 3

En esta última actividad se podrá volver a la anterior mediante la flecha ubicada en la parte inferior izquierda de la pantalla o se podrá aceptar para dar por acabada la actividad y regresar a la pantalla inicial de la aplicación.

4.2 DISEÑO INTERNO

En esta sección se explicarán brevemente las diferentes alternativas de diseño que se han tenido en cuenta durante el desarrollo de la aplicación.

4.2.1 ALTERNATIVAS DISEÑO INTERNO

En cuanto al diseño interno de la aplicación también se han barajado diferentes posibilidades de desarrollo desde el inicio de la aplicación. En concreto se barajaron tres diferentes posibilidades que serán detalladas a continuación.

4.2.1.1 DISEÑO SIN BASE DE DATOS

Inicialmente, el diseño se pensó realizar sin añadir ninguna base de datos. Esto se pensó así ya que, para la funcionalidad de activar los recordatorios, no se creía conveniente utilizarla, puesto que se pensaba desarrollar mediante el calendario que traen los propios dispositivos Android. Es decir, los recordatorios serían guardados directamente sobre este calendario el cual se encargaría de lanzar los mismos en las fechas oportunas.

Más adelante, las ideas que surgen durante el desarrollo hicieron que los pensamientos cambiaran, y por lo tanto cambiara la forma en la que se desarrollaría la funcionalidad de los recordatorios tal y como se verá en el siguiente apartado.

4.2.1.2 DISEÑO CON BASE DE DATOS REMOTA

Durante el desarrollo de la aplicación, fueron surgiendo cambios en la misma. Entre estos cambios se encontraba el de cambiar la forma de revisar cada una de las partes del vehículo. En vez de mostrar información sobre cada una de estas partes del vehículo se pensó que era más entretenido y de bastante más utilidad para el usuario, ya que de esta forma iría revisando cada una de las partes y marcando su estado como correcto o incorrecto para poder verlo en cualquier momento y poder recordarlo de esta manera nada más verlo.

Es por esto que surgió la necesidad de incorporar una base de datos, ya que era necesario almacenar el estado de cada una de las partes del vehículo a revisar, para poder cargarlo cada vez que el usuario accediera a esta funcionalidad.

Por otro lado, y aprovechando la incorporación de la base de datos, se llegó a la conclusión que los recordatorios también serían almacenados en la misma, ya que de esta manera sería mucho más fácil recuperar los datos de los mismos para poder trabajar con ellos.

Inicialmente se consideró como mejor opción trabajar con una base de datos remota alojada en un servidor dedicado para ello. Esta opción finalmente se descartó, ya que contaba con diversos inconvenientes, como por ejemplo la búsqueda del servidor que se utilizaría, o la velocidad de acceso a los datos, que sería más lenta que si se hiciera de forma local.

4.2.1.3 DISEÑO CON BASE DE DATOS NO REMOTA

Finalmente se llegó a la conclusión de que la mejor opción sin duda era el uso y generación de una base de datos local que se puede utilizar en cualquier dispositivo Android.

Con esto se conseguiría muchas mejoras, como la facilidad y rapidez en el acceso a los datos almacenados en la propia base de datos creada en el dispositivo.

De esta forma, se generará la base de datos con su modelo relacional correspondiente que será detallado más adelante y paliando de esta manera la funcionalidad de los recordatorios y la forma de guardar y recuperar el estado de cada una de las partes que hay que revisar de cada vehículo.

4.2.2 DIAGRAMA DE COMPONENTES

En este apartado se detallará el diagrama de componentes de la aplicación, el cual servirá para ver los diferentes componentes con los que cuenta la aplicación y las dependencias entre ellos.

En el diagrama que se puede observar a continuación, Figura 36, se pueden observar los siguientes componentes:

- **Model:** Contiene todas las clases que componen el modelo de datos y que se utilizará en los demás componentes.
- **Controller:** Este componente poseerá las clases que se encargarán de controlar el intercambio de datos entre las listas y la base de datos.
- **Lists:** Este componente es el encargado del manejo de todas las listas que se utilizarán para mostrar toda la información relevante.
- **Views:** Componente que se encargará de controlar la forma en que se muestran las vistas en las diferentes pantallas.

- **BBDD:** Módulo que se encargará del manejo de datos que reciba y que posteriormente se utilizarán.

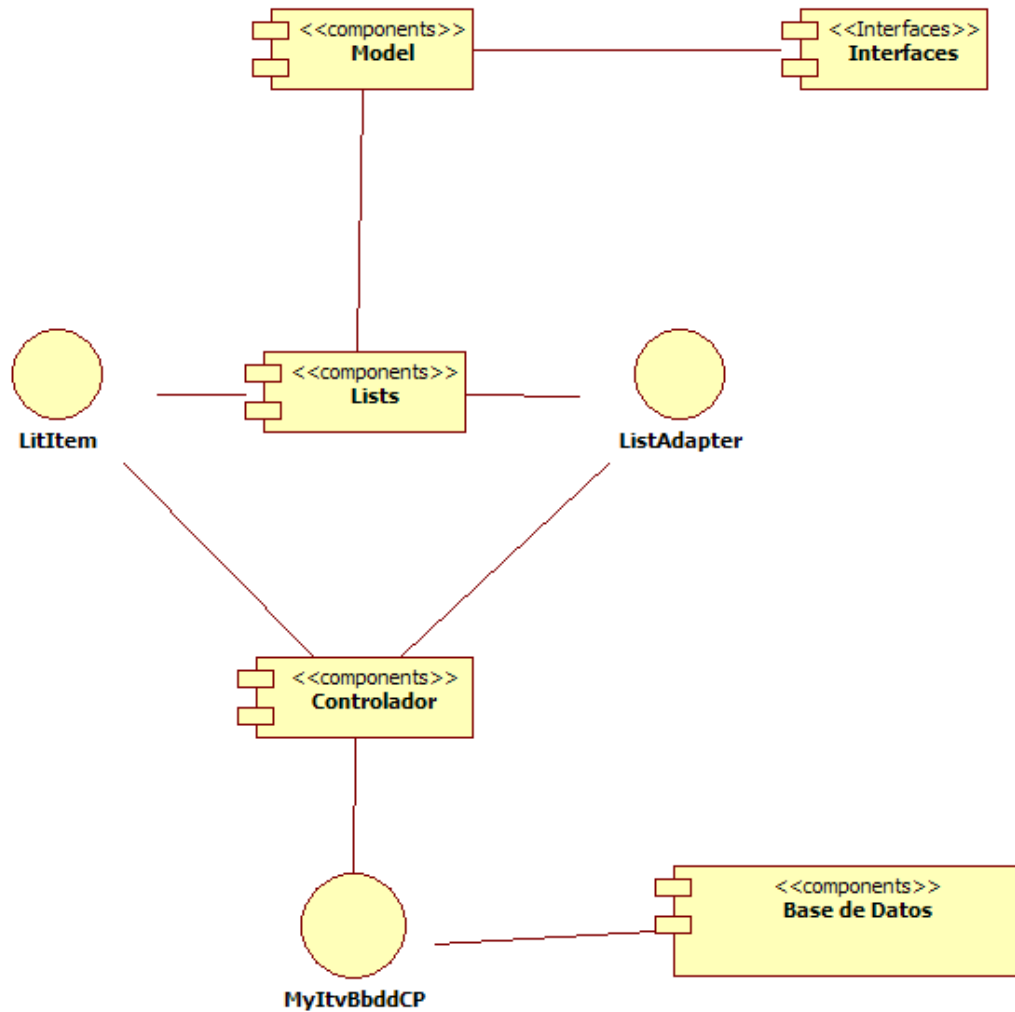


Figura 36: Diagrama de componentes de la aplicación

4.2.3 DIAGRAMA DE SECUENCIA

Tras el diseño del diagrama de componentes se procederá a definir los diagramas de secuencias que servirán para ver la relación entre los diferentes objetos

de la aplicación a lo largo del tiempo y mostrando la secuencia de mensajes intercambiados entre ellos.

El primer diagrama es el que se puede observar en la Figura 37. La clase *MyItvActivity* se inicializa junto a la aplicación y muestra el menú principal de la misma.

En ella aparecen cuatro opciones. En este primer diagrama y debido al gran número de clases que existen sólo se muestra la secuencia con la opción de *¿Qué Revisar?*, y al elegir esta opción, como se puede observar en dicho diagrama, se carga un dialogo mediante el método *dialog.show()* mediante el cual se podrá elegir el tipo de vehículo sobre el que se desea realizar la revisión de sus partes. En este primer caso se muestra la secuencia primero del *Alumbrado* para el coche. Como se puede observar, al pulsar sobre la opción de *Alumbrado* lo primero que se produce es una llamada a la clase *ConfigurarListView*. Esta clase se encargará de inflar la lista mediante el método *inflate()*. Una vez inflada la lista con todos sus elementos, se podrá pulsar en cualquiera de ellos. Al ser pulsado cada uno de ellos se accederá a la siguiente clase. Por comodidad, a la hora de realizar el diagrama solo se han incluido tres de los diez elementos que existirán en la lista de *Alumbrado*. Estando en cualquiera de ellos, por ejemplo en la clase *LucesPosicion*, se podrá acceder a la siguiente clase, en este caso *LucesPosicion2*, y una vez en esta clase y dependiendo de la respuesta que se genere se volverá a la clase *Alumbrado* en la cual se volverá a inflar la lista mediante el método *inflate()*.

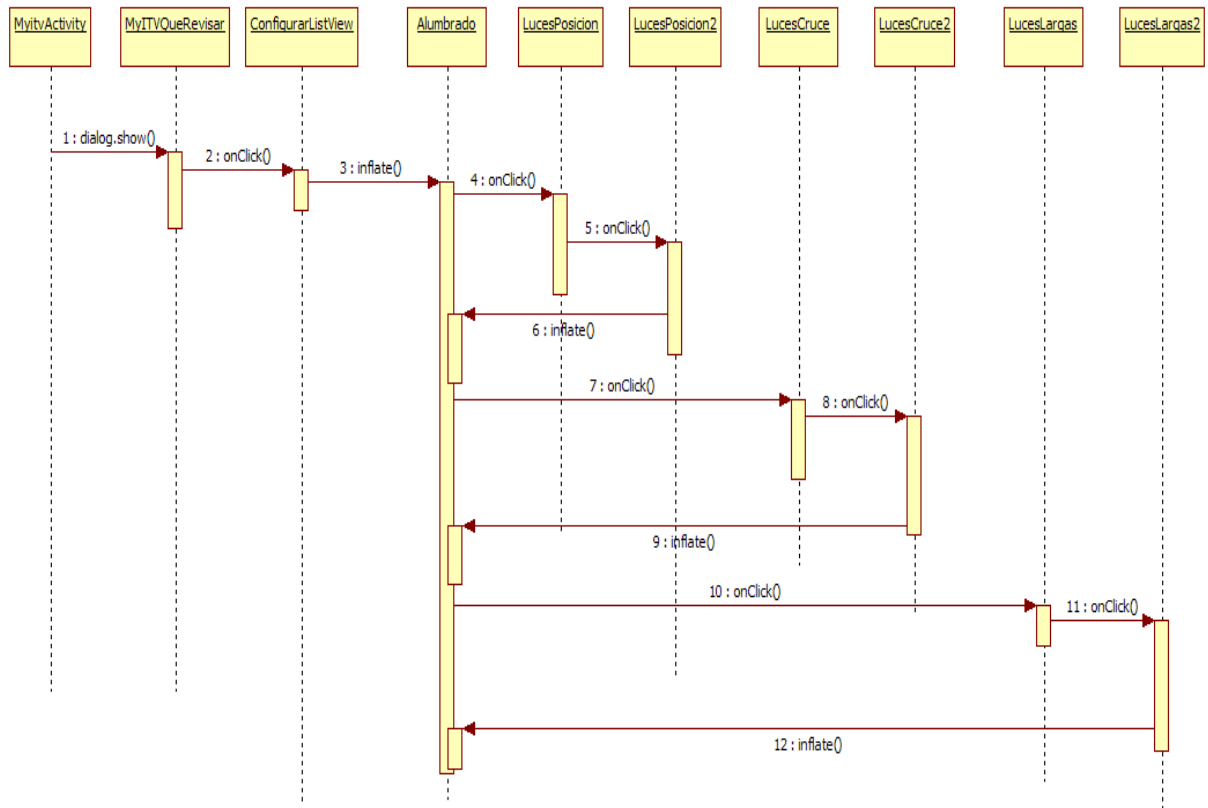


Figura 37: Diagrama de secuencia revisar alumbrado coche

El siguiente diagrama es el que hace referencia a la clase *Neumaticos*, también para la opción de coche, la cual es bastante similar a la anterior. Dicho diagrama es el que se puede observar en la Figura 38. Al igual que en el anterior caso se comienza en la actividad principal *MyItvActivity*. Seguidamente se accederá a la clase *MyITVQueRevisar*, y dentro de esta se pulsará la opción *Neumaticos*. Antes de acceder a la clase *Neumaticos* se accederá a la clase *ConfigurarListViewNeumaticos*, la cual se encargará de inflar la lista mediante el método *inflate()*. Una vez inflada la lista ya se podrá seleccionar cualquiera de sus elementos, accediendo así a cada una de las diferentes clases. En el caso de acceder a la clase *NeumaticosHomologados*, desde esta clase se podrá continuar a la siguiente que es *NeumaticosHomologados2*. Estando en esta última clase y dependiendo de la respuesta elegida se procederá a volver a inflar la lista de la clase *Neumaticos* mediante el método *inflate()*.

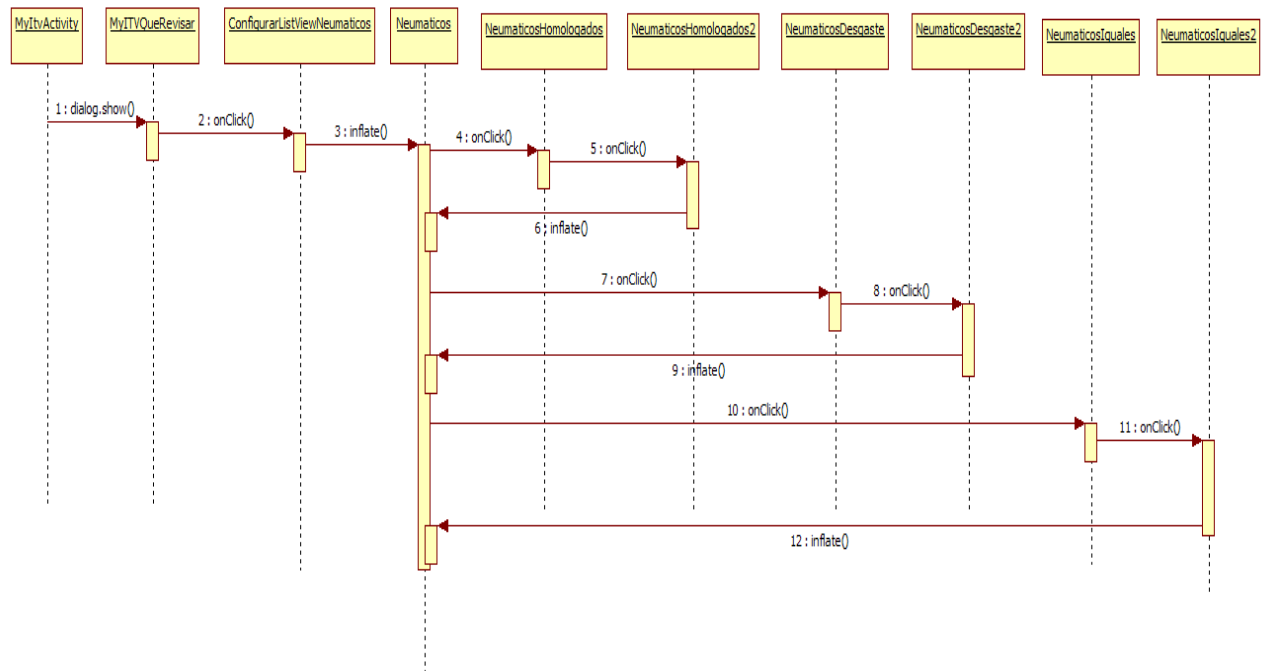


Figura 38: Diagrama de secuencia revisar neumáticos coche

De forma muy similar, se obtendrá el siguiente diagrama de secuencia correspondiente a la opción de “Frenos” elegida dentro de “¿Qué Revisar?” y con la opción de coche y es el que se puede observar en la Figura 39. Al igual que en los diagramas anteriores se comenzará en la clase principal *MyItvActivity*. Una vez en ella se elegirá nuevamente la opción “¿Qué Revisar?”, la cual nos llevará a la clase *MyITVQueRevisar* y nos volverá a mostrar las cuatro opciones disponibles. Accederemos a la opción *Frenos* y accederemos a la clase *ConfigurarListViewFrenos* la cual se encargará de inflar la lista para los elementos de los frenos mediante el método *inflate()*. Una vez inflada la lista se accederá directamente a la clase *Frenos*. En dicha clase se podrá seleccionar el elemento de la lista que se desee. Se accede, por ejemplo, a la clase *FrenoVibracion* desde la cual se accederá seguidamente a la clase *FrenoVibracion2*. Estando en esta última y dependiendo de la respuesta elegida, se volverá a inflar la lista con los nuevos resultados mediante el método *inflate()*.

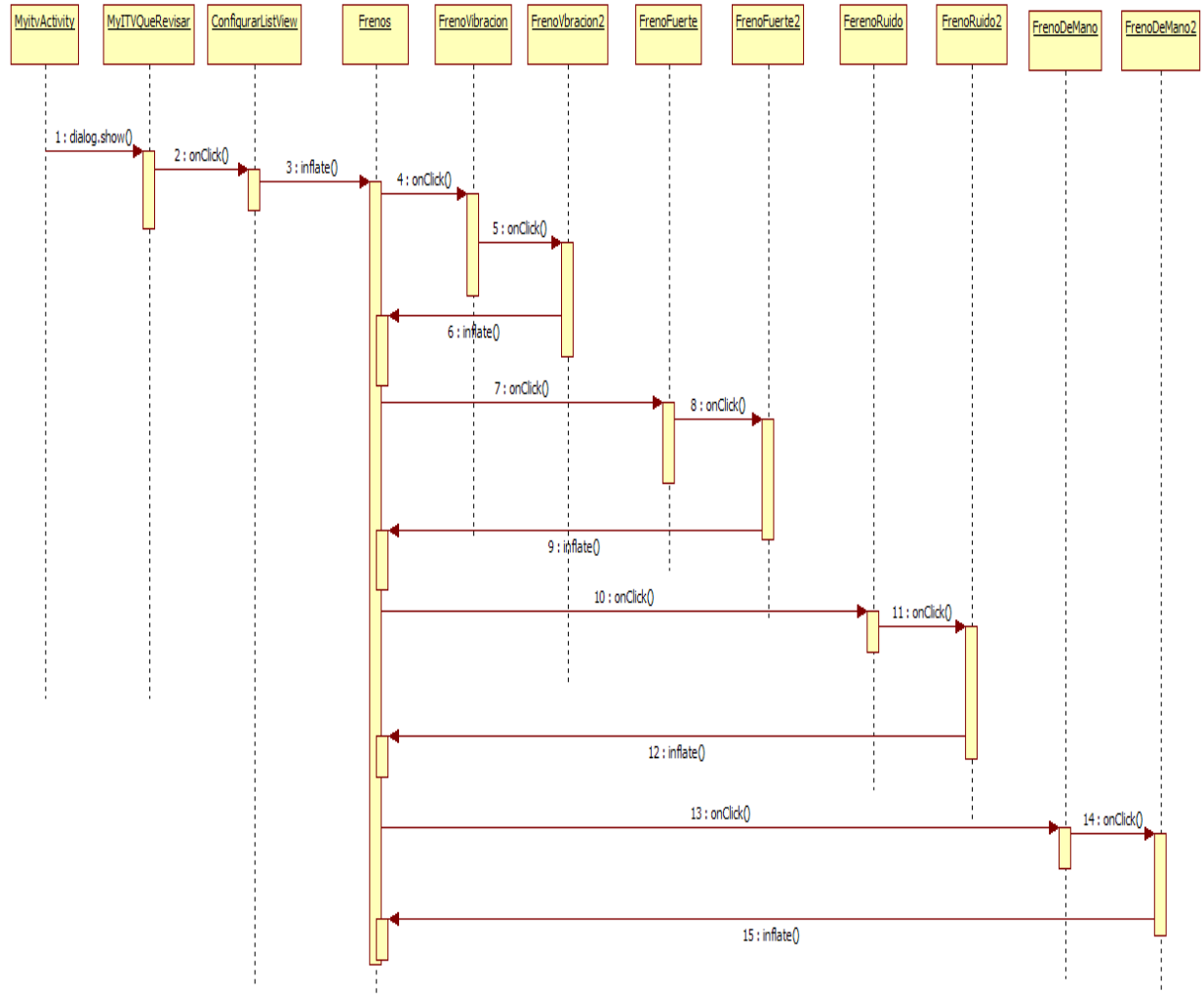


Figura 39: Diagrama de secuencia revisar frenos coche

El siguiente diagrama es el que hace referencia a la clase *General*, la cual es bastante similar a las anteriores y correspondiente también a la opción de coche. Dicho diagrama es el que se puede observar en la Figura 40. Se comienza nuevamente en la actividad principal *MyItvActivity*. Seguidamente se accederá a la clase *MyITVQueRevisar*, y dentro de esta se elegirá la opción *General*. Antes de acceder a la clase *General* se accederá a la clase *ConfigurarListViewGeneral*, la cual se encargará de inflar la lista mediante el método *inflate()*. Una vez que dicha lista se encuentre inflada ya se podrá seleccionar cualquiera de sus elementos, accediendo así a cada una de las diferentes clases. En el caso de acceder a la clase *GeneralAceite*, desde ésta se podrá continuar a la siguiente que es *GeneralAceite2*. Estando en esta última clase y dependiendo de la respuesta elegida se procederá a volver a inflar la lista de la clase *General* mediante el método *inflate()*.

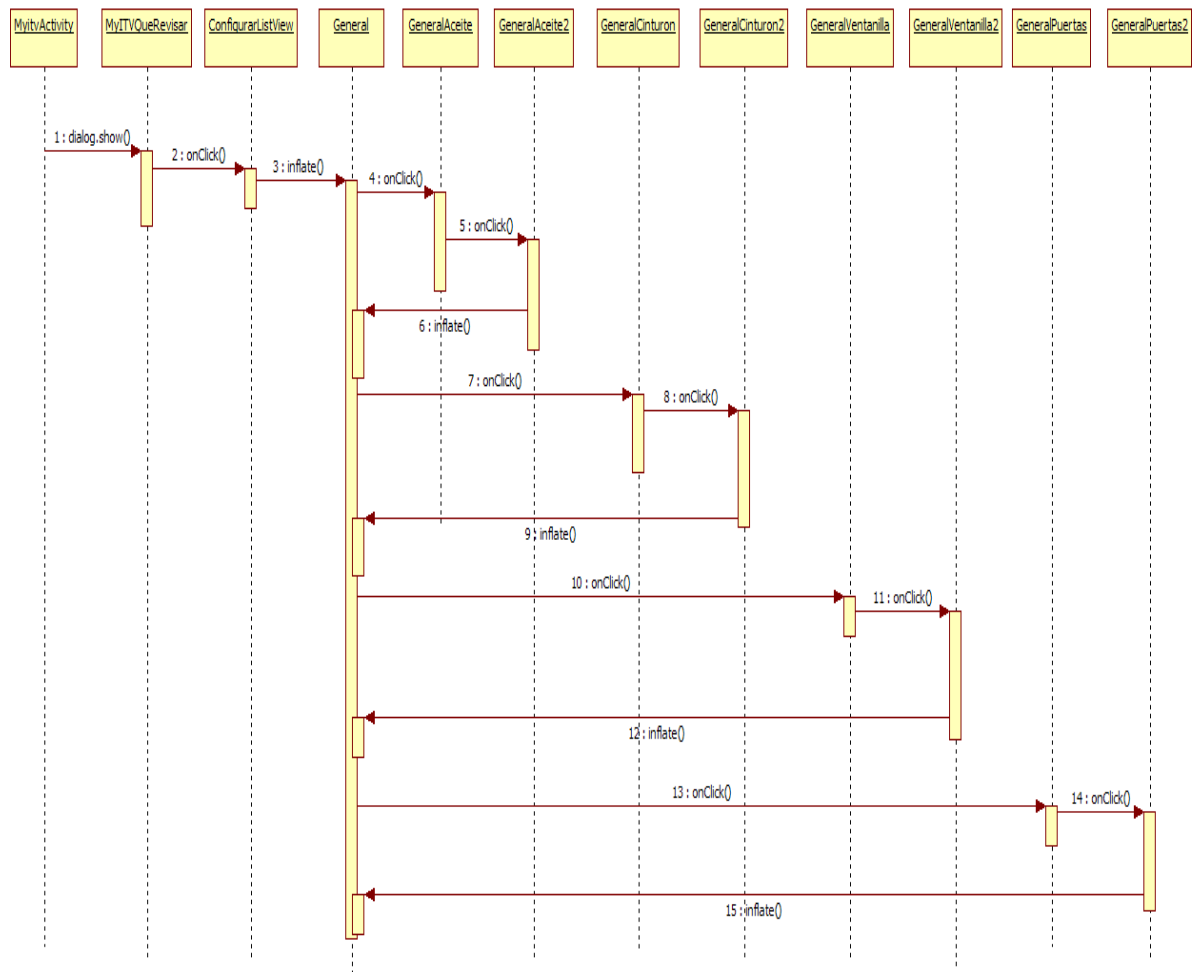


Figura 40: Diagrama de secuencia revisar general coche

Los siguientes diagramas son los que hacen referencia a las partes que revisar de las motocicletas. Dicha opción es elegida igual que en el caso anterior mediante el método *dialog.show()* de la clase *MyItvActivity*.

El primer diagrama para esta opción es el que se puede observar en la Figura 41 y que hace referencia a la funcionalidad de revisar el alumbrado de la motocicleta. Como se puede observar el proceso de transición entre estas clases es idéntico al mostrado en los diagramas anteriores.

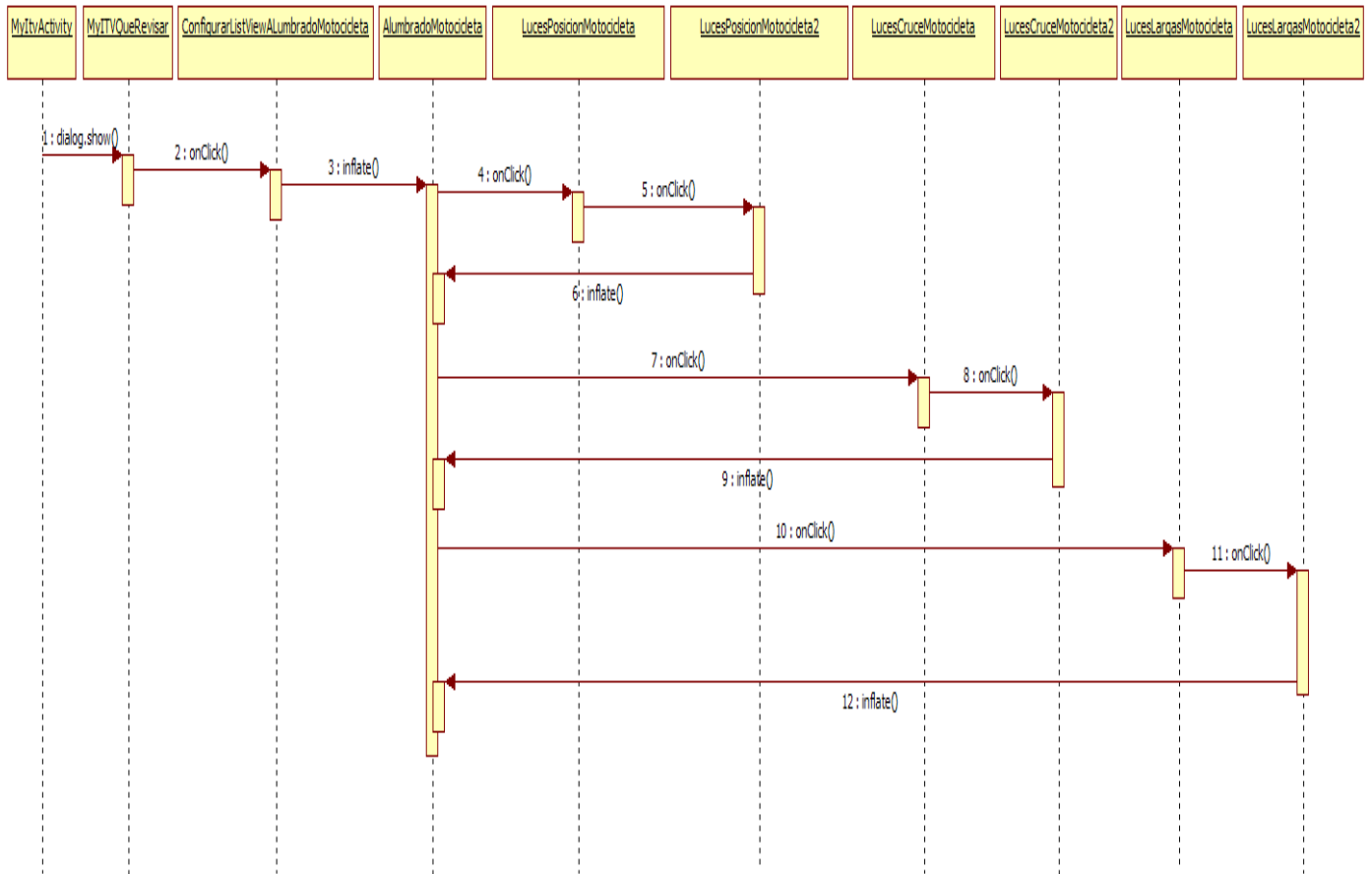


Figura 41: Diagrama de secuencia revisar alumbrado motocicleta

El siguiente diagrama, Figura 42 muestra las secuencia de clases que se sigue para la funcionalidad de revisar los neumáticos de la motocicleta y que como se puede observar sigue los mismos pasos que los anteriores diagramas.

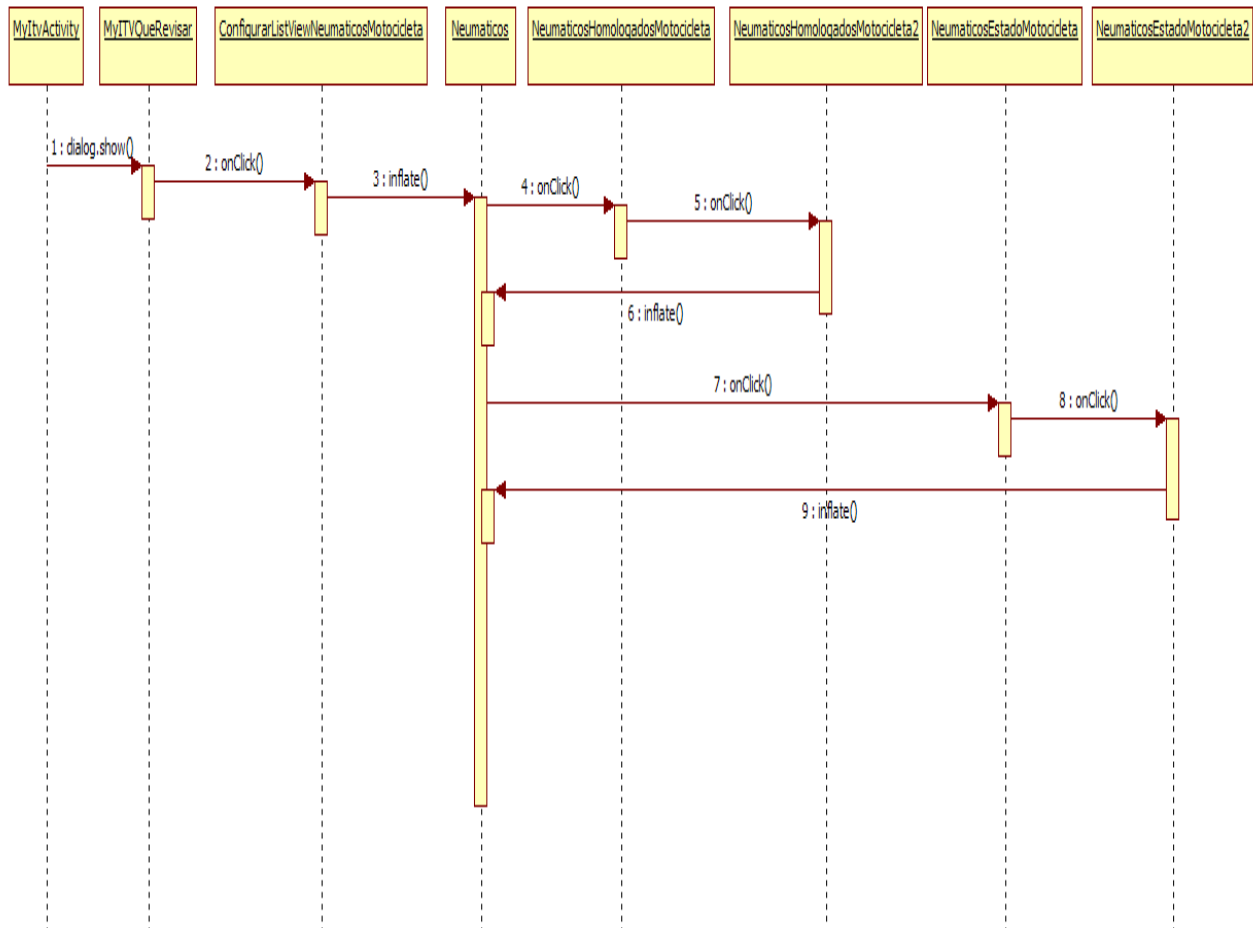


Figura 42: Diagrama de secuencia revisar neumáticos motocicleta

El siguiente diagrama es el que se corresponde con las clases para la funcionalidad de revisar los frenos de la motocicleta. Como se puede observar los pasos que se siguen son idénticos que en los casos anteriores ya que la funcionalidad es la misma.

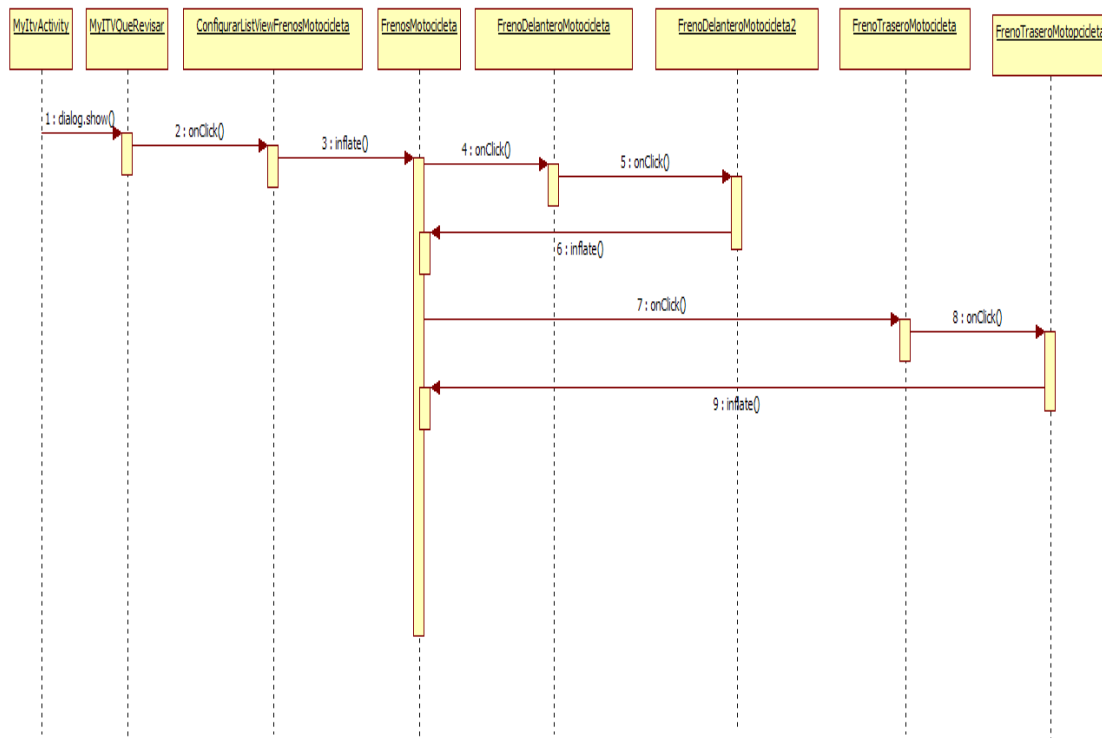


Figura 43: Diagrama de secuencia revisar frenos motocicleta

El último diagrama perteneciente a la revisión de la motocicleta es el que se puede observar en la Figura 44. Como se puede ver el procedimiento que se sigue para la secuencia de estas clases es idéntico al visto en los diagramas anteriores. Al igual que en el caso del alumbrado en este diagrama también se han suprimido algunas clases ya que por su extensión no se creía necesario añadirlas puesto que la secuencia entre las clases sería idéntica.

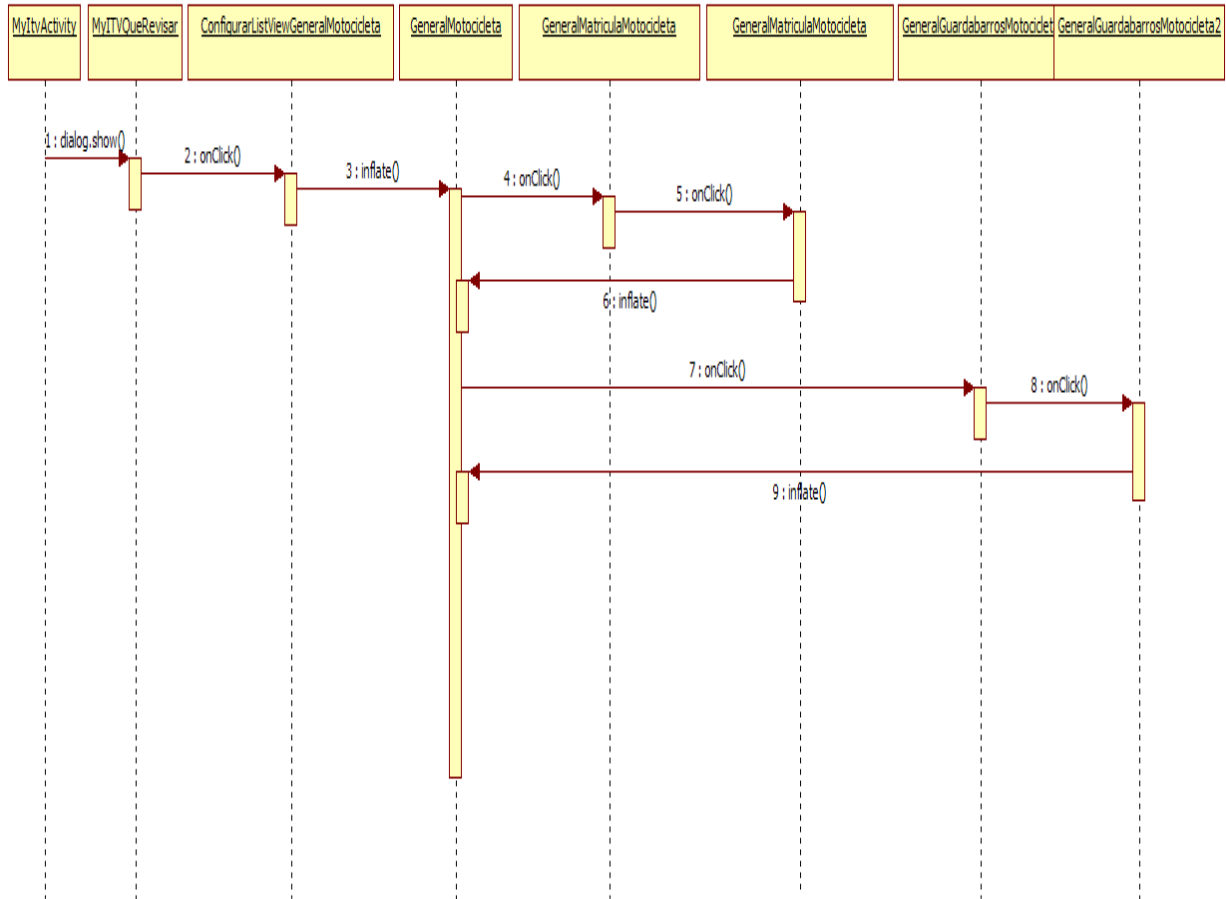


Figura 44: Diagrama de secuencia revisar general motocicleta

El último diagrama de secuencia Figura 45 muestra las tres últimas opciones que se pueden elegir desde la clase principal *MyItvActivity*. Si la clase a la que se accede es *MyITVProximaRevision* en ella podremos calcular la próxima inspección del vehículo. Una vez pulsado el botón para calcular se accederá a la siguiente clase *ConfirmarRevision* en la cual se podrá activar un recordatorio. Si se elige la opción para activar dicho recordatorio aparecerá un dialogo mediante el método *dialogo.show()*. Este dialogo se mostrará mediante la clase *DialogoPersonalizado* y en él se introducirán los datos para la notificación. Si finalmente se acepta la notificación ésta será activada en la clase *AlarmReceiver* mediante el método *alarm.set()* y seremos

devueltos a la clase principal de la aplicación *MyItvActivity*. Si por el contrario no se quisiera finalmente aceptar la notificación el usuario será devuelto nuevamente a la clase *MyITVProximaRevision*. Por otro lado, en esta misma clase *MyITVProximaRevision* también se podrá acceder a la clase *InfoRevisiones* en la cual se ofrecerá la información sobre los periodos que deberá pasar cada vehículo la inspección oportuna.

Si la desde la clase principal de la aplicación se accede a la clase *MisNotificaciones* en ella se podrán ver todas las notificaciones almacenadas hasta el momento, y se dará la opción de poder eliminarlas mediante el método *delete()* manteniéndose el usuario en esta misma clase.

Por último si la opción que se elige desde la actividad principal es ir a la clase *DocumentacionPermisoCirculacion* en ella se podrá ver el primer documento necesario. Avanzando a la siguiente clase *DocumentacionFichaTecnica* se verá el siguiente documento necesario, y por último se podrá acceder a la última clase *DocumentacionSeguro* en el que se podrá ver el último documento necesario.

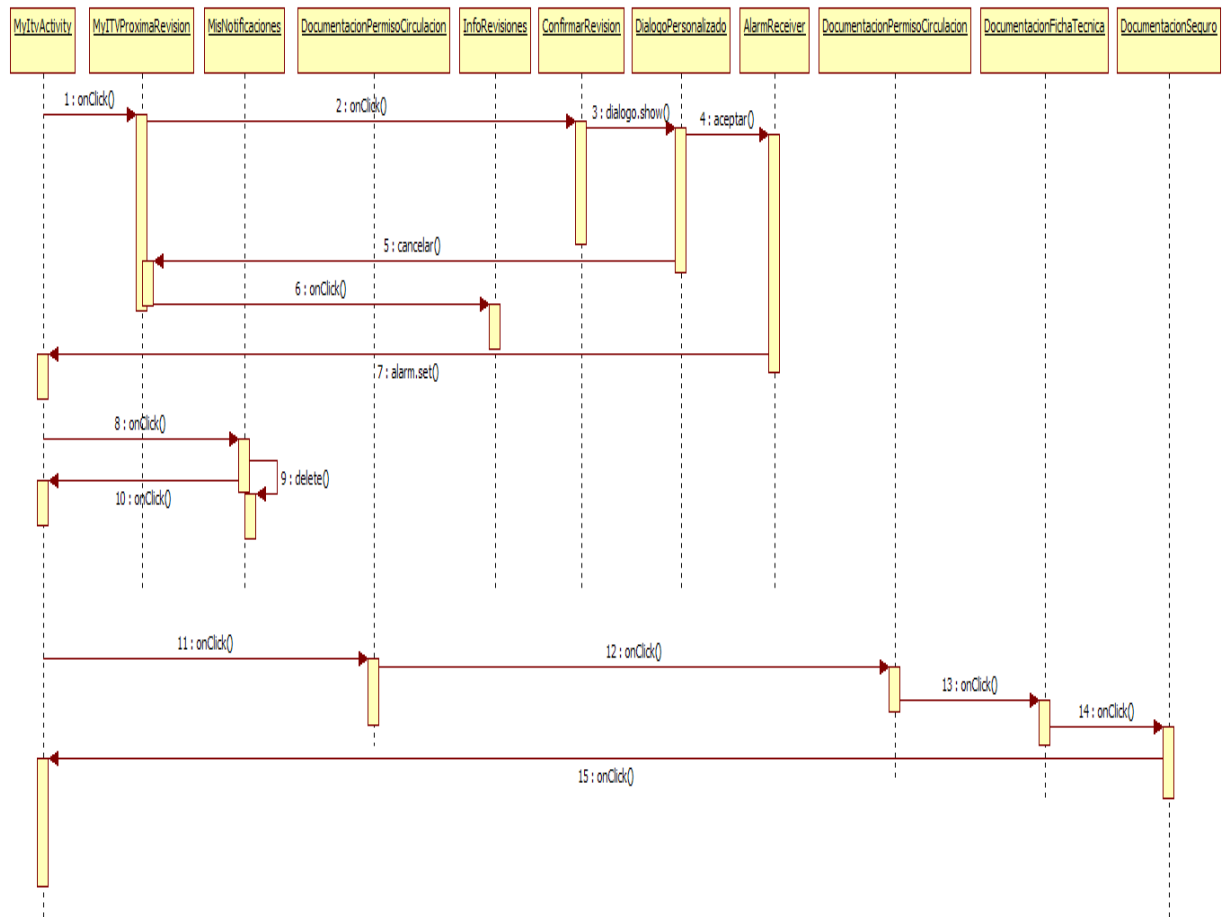


Figura 45: Diagrama de secuencia documentación y notificaciones

4.2.4 DIAGRAMA DE CLASES

En este apartado se detallará el diagrama de clases, el cual servirá para visualizar las relaciones entre las clases que se involucran en el sistema.

En la Figura 46 aparecen las cinco primeras clases. Se ha decidido separar los diagramas en varios, ya que, debido al gran número de clases existentes en la aplicación, se cree conveniente realizarlo por partes, para así tener una mejor visión del mismo.

Inicialmente se muestra la clase principal de la aplicación, *MyItvActivity* en la cual se mostrarán las cuatro opciones principales que el usuario puede elegir. El acceso a las siguientes clases es implementado mediante “*Intents*”, en el cual se le pasará por parámetros el contexto de la clase actual y la clase a la que se quiere acceder. Una vez generado el “*Intent*”, se iniciará la siguiente actividad mediante el método

startActivity(). Además, en esta misma clase, también se encuentra el método *manejadorOpciones()*, que servirá para controlar la opción que el usuario elija en la pantalla. Este método será implementado en muchas de las siguientes clases y tendrá la misma funcionalidad en todas.

La primera opción será la que se implementa mediante la clase *MyITVQueRevisar*. Como se puede observar en esta clase se cuenta con un método *crearDialogoSeleccion()* que será el encargado de generar un diálogo en el que se le dará al usuario la opción de elegir entre revisar el coche o revisar la motocicleta. Dependiendo de la opción que se elija se accederá a un conjunto de clases o a otro, tal y como se puede ver en la Figura 48 y Figura 49.

La clase *MyItvProximaRevision* es la que se encarga de calcular la fecha para la próxima inspección mediante el tipo y la fecha de matriculación del vehículo.

La clase *MisNotificaciones* es la encargada de mostrar al usuario todas las notificaciones que se hayan almacenado hasta el momento. Además esta misma clase posee el método *crearDialogoAlerta(View)* que servirá para lanzar un dialogo cuando el usuario intente eliminar una notificación para avisarle de que se va a proceder a ello.

La clase *DocumentacionPermisoCirculacion* es una clase que sirve para mostrar la documentación necesaria para la inspección del vehículo. En este caso mostrará como documentación el permiso de Circulación.

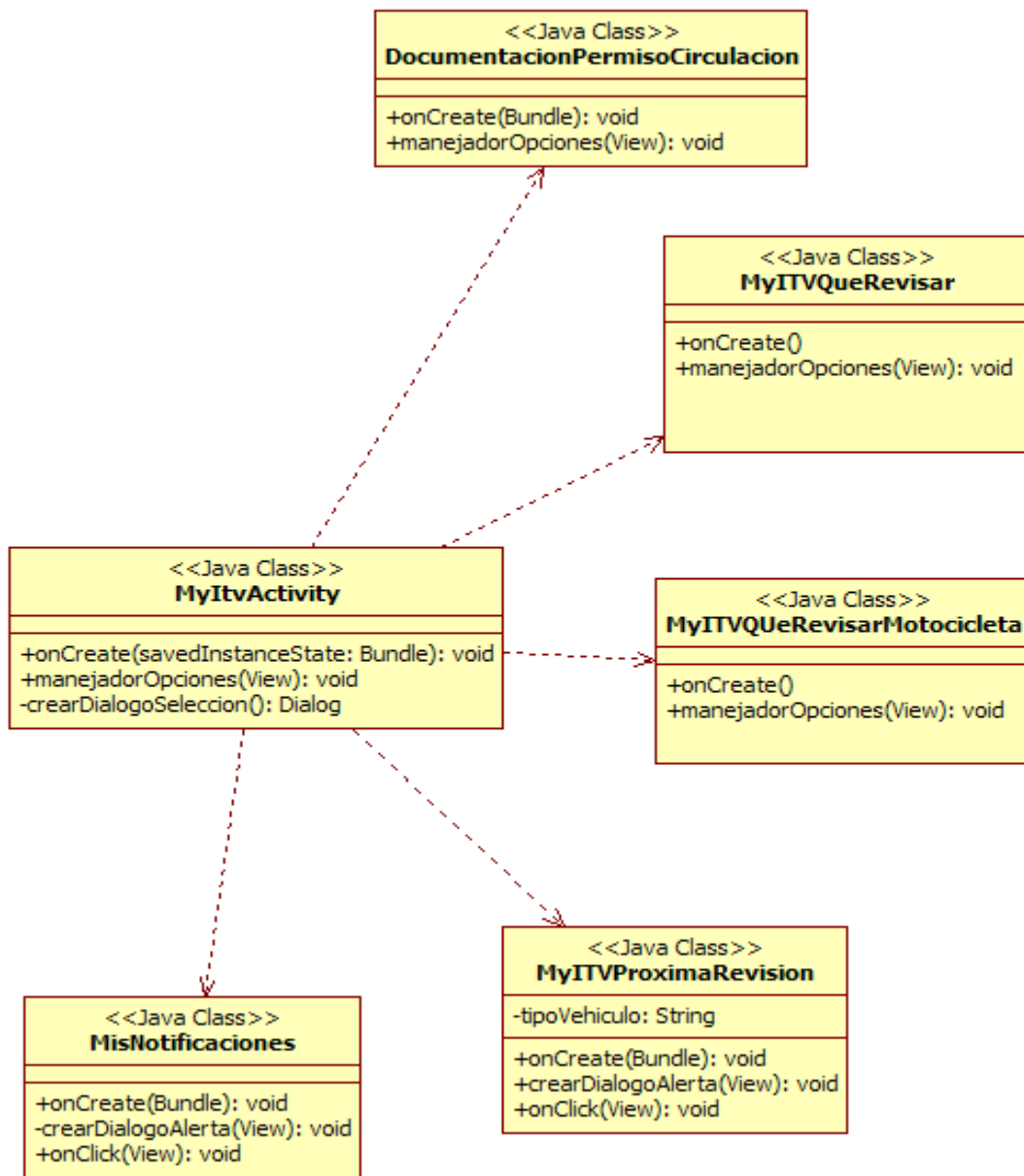


Figura 46: Diagrama de clases actividad principal

En la Figura 47 se muestra una sola clase independiente. Esta clase *ActionBar* es la encargada de mostrar por pantalla la barra de acciones personalizada durante toda la aplicación.

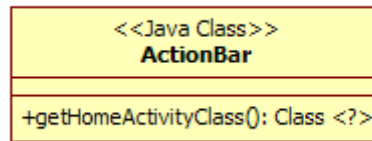


Figura 47: Diagrama de clases actionBar

En la Figura 48 se muestran las siguientes clases de la aplicación. A estas clases se accede desde la clase *MyITVQueRevisar*, dependiendo de la opción elegida en el dialogo de selección tal y como se comentó anteriormente.

Por un lado se podrá acceder a las partes que hay que revisar para un coche Figura 48, a las que se accede desde la clase *MyITVQueRevisar*, y por otro lado a las partes que hay que revisar para una motocicleta Figura 49 a las cuales se accede desde la clase *MyITVQueRevisarMotocicleta*.

Como se puede observar, estas clases son casi idénticas debido a que todas ellas poseen la misma funcionalidad. Por ello sólo se procederá a explicar solamente una de ellas.

Por ejemplo, la clase *Neumaticos* tiene como principal objetivo cargar una lista con todos los elementos que es necesario revisar de los neumáticos del vehículo. Para ello hará uso de la clase *ConfigurarListViewNeumaticos*, la cual se encargará de inflar la lista con sus respectivas imágenes. Con cada elemento de la lista se accederá a otra clase diferente que serán explicadas a continuación. Todas las demás clases de la Figura 48 funcionan de manera idéntica.

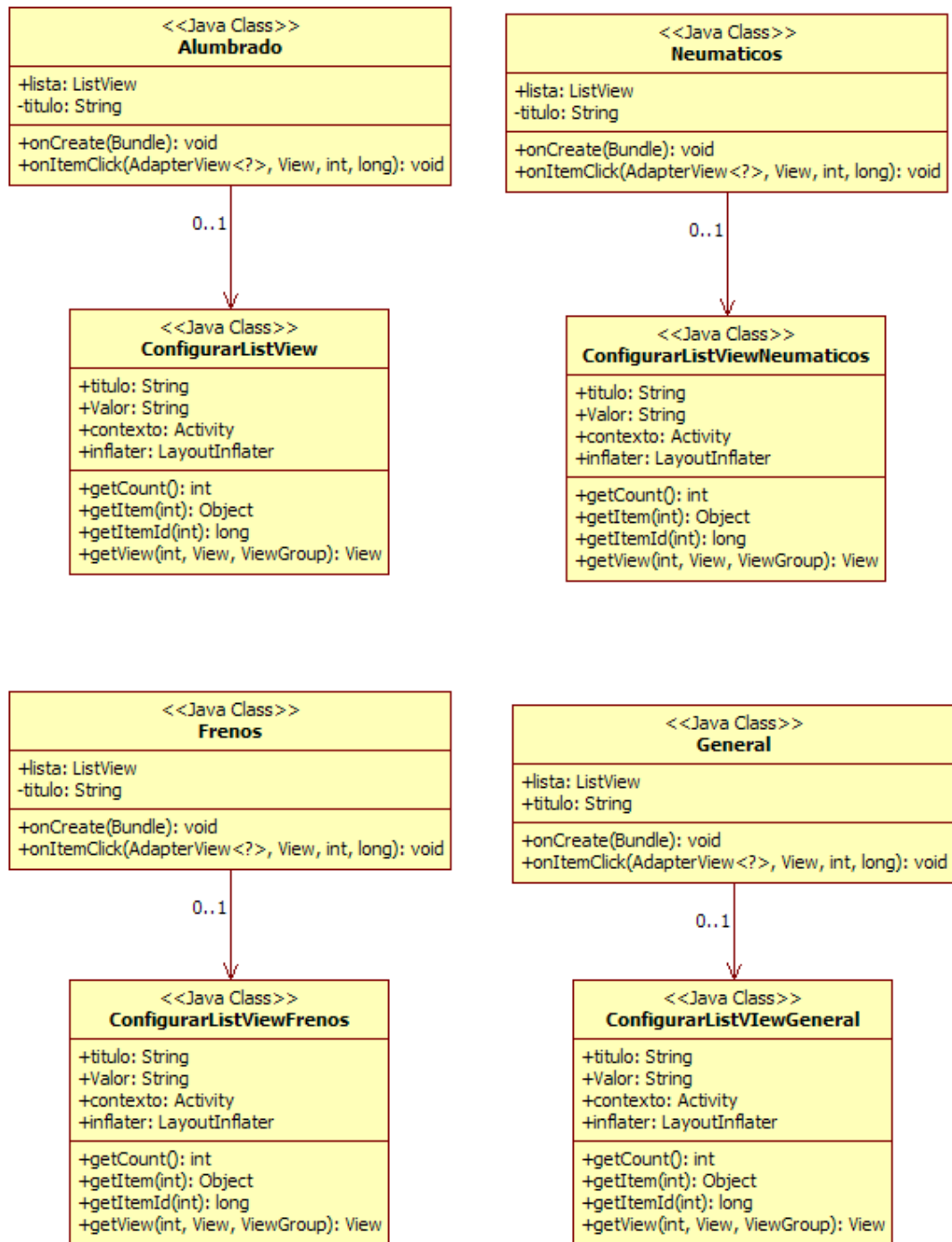


Figura 48: Diagrama de clases que revisar coche

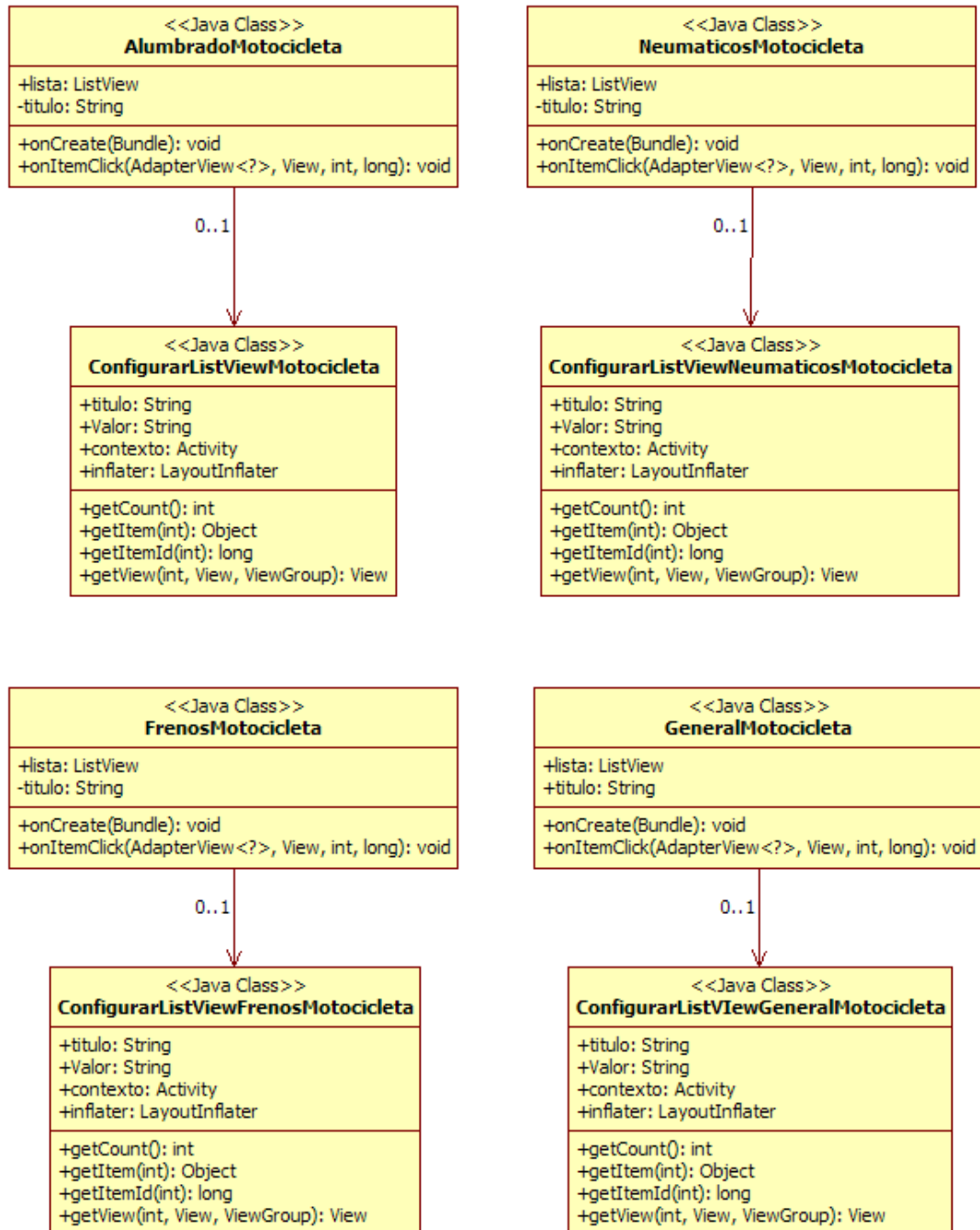


Figura 49: Diagrama de clases que revisar motocicleta

La Figura 50 muestra las clases que se cargarían cuando el usuario elija la opción de revisar el alumbrado del vehículo.

A todas estas clases se accede desde la clase *Alumbrado*, por lo cual, todas guardarán una relación de dependencia con ella. Como se puede observar, las clases *LucesDireccion*, *LucesDireccion2* y *LucesEmergencia* tienen anidadas una clase *Animator* la cual servirá para llevar a cabo las animaciones [9] que mostrarán por pantalla.

Una posible transición sería desde la clase *Alumbrado* ir hasta la clase *LucesPosicion*, en la cual se mostrará la información necesaria al usuario y desde ésta acceder a la clase *LucesPosicion2* en la cual se formulará una pregunta al usuario y dependiendo de la respuesta del mismo la lista se actualizará de una forma o de otra.

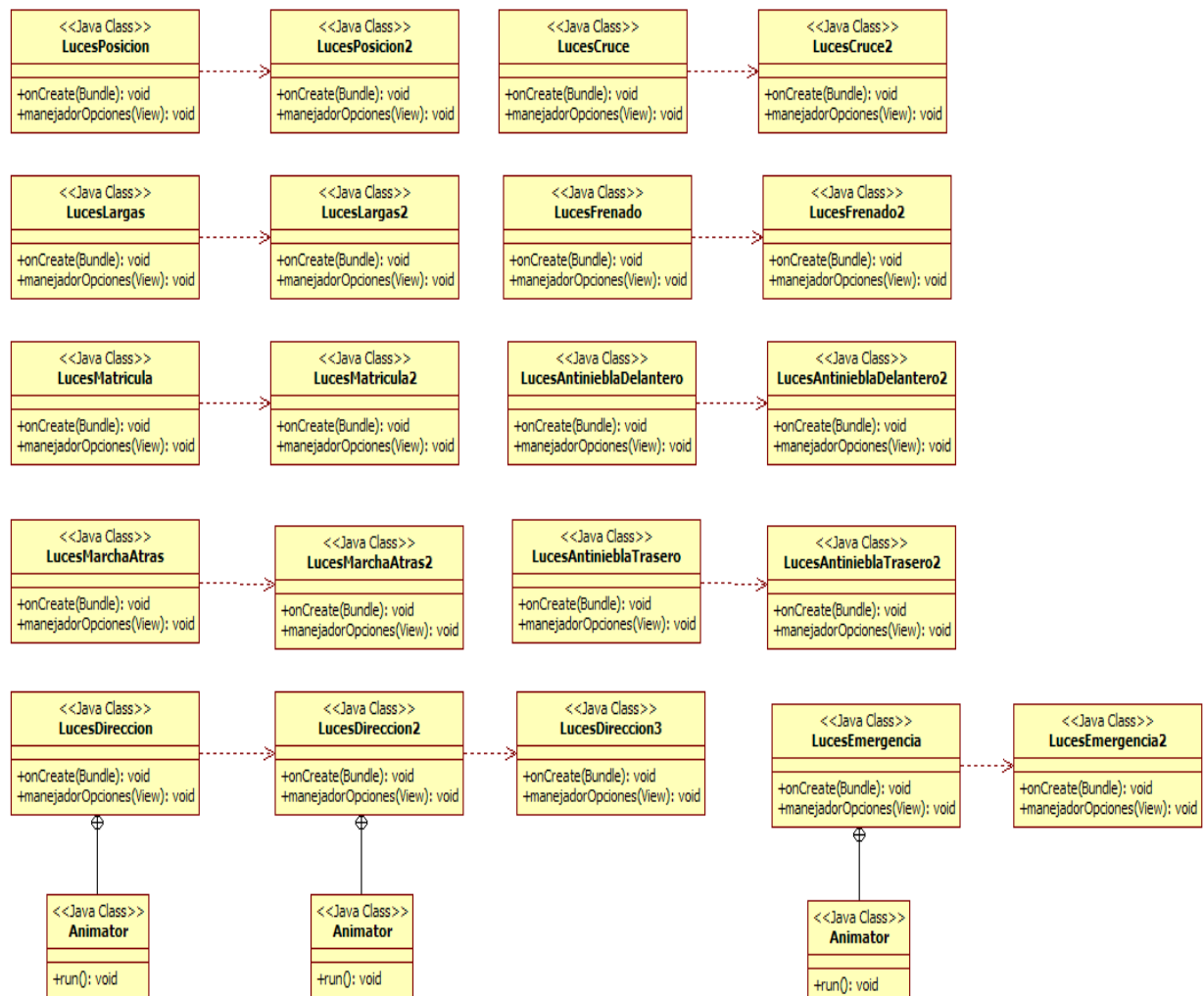


Figura 50: Diagrama de clases revisar alumbrado coche

El siguiente diagrama Figura 51 se corresponde con las clases relacionadas con la opción de frenos para el coche. A todas estas clases se accede desde la clase *Frenos*, es decir, desde la lista que se genera en dicha clase se accederá a cada una de las que se puede observar en el diagrama. En dicho diagrama se observar también la dependencia que existe entre las diferentes clases, que al igual que en el caso anterior tienen la misma funcionalidad.

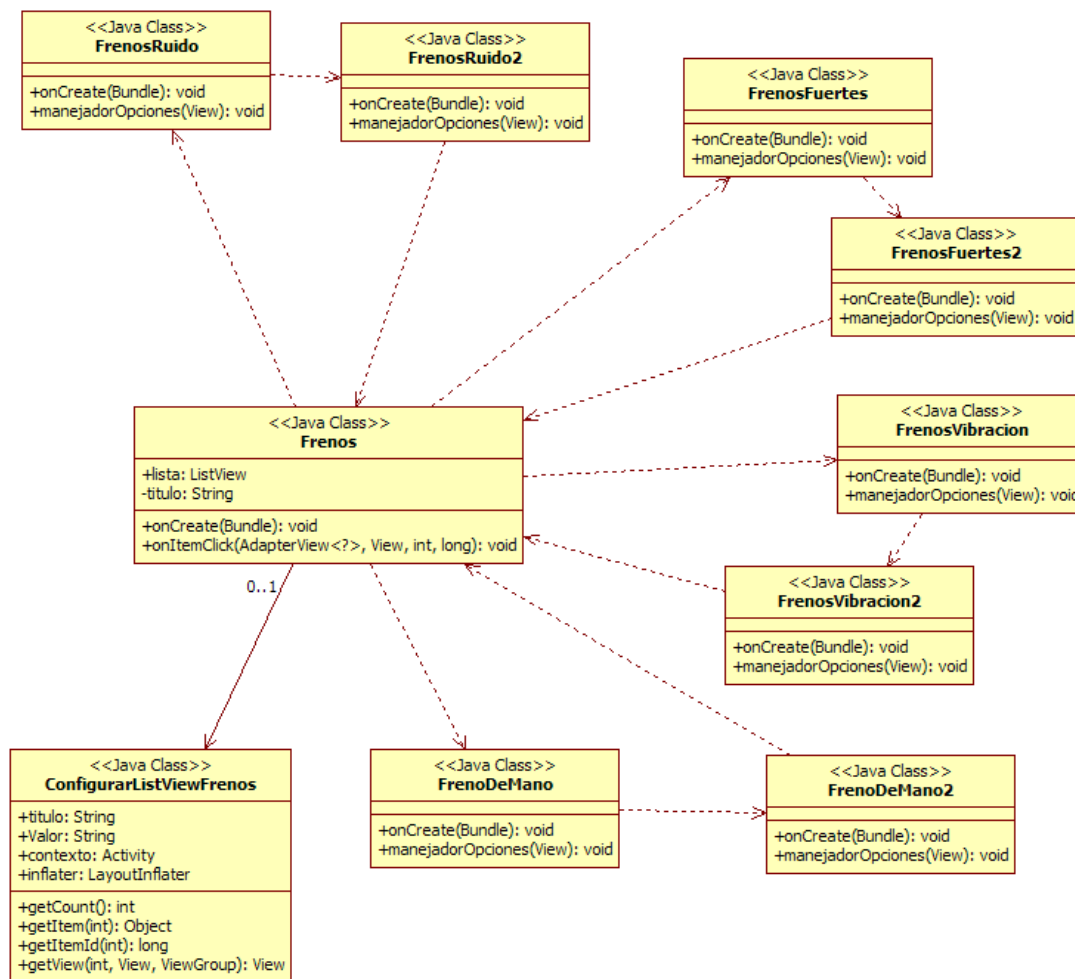


Figura 51: Diagrama de clases revisar frenos coche

El siguiente diagrama, Figura 52, es el que se corresponde a las clases que se lanzan cuando el usuario selecciona la opción de revisar neumáticos para la opción de coche. Como se puede observar a estas clases se accede desde la clase *Neumaticos* y estas muestran una funcionalidad idéntica a las vistas anteriormente. Desde la lista generada en la clase *Neumaticos* se accede, por ejemplo, a la clase *NeumaticosDesgaste* en la cual se muestra información al usuario. Desde dicha clase se

accederá a la siguiente, *NeumaticosDesgaste2* en la cual se formulará una pregunta al usuario y, dependiendo de su respuesta se volverá a cargar la lista de los neumáticos con un valor u otro.

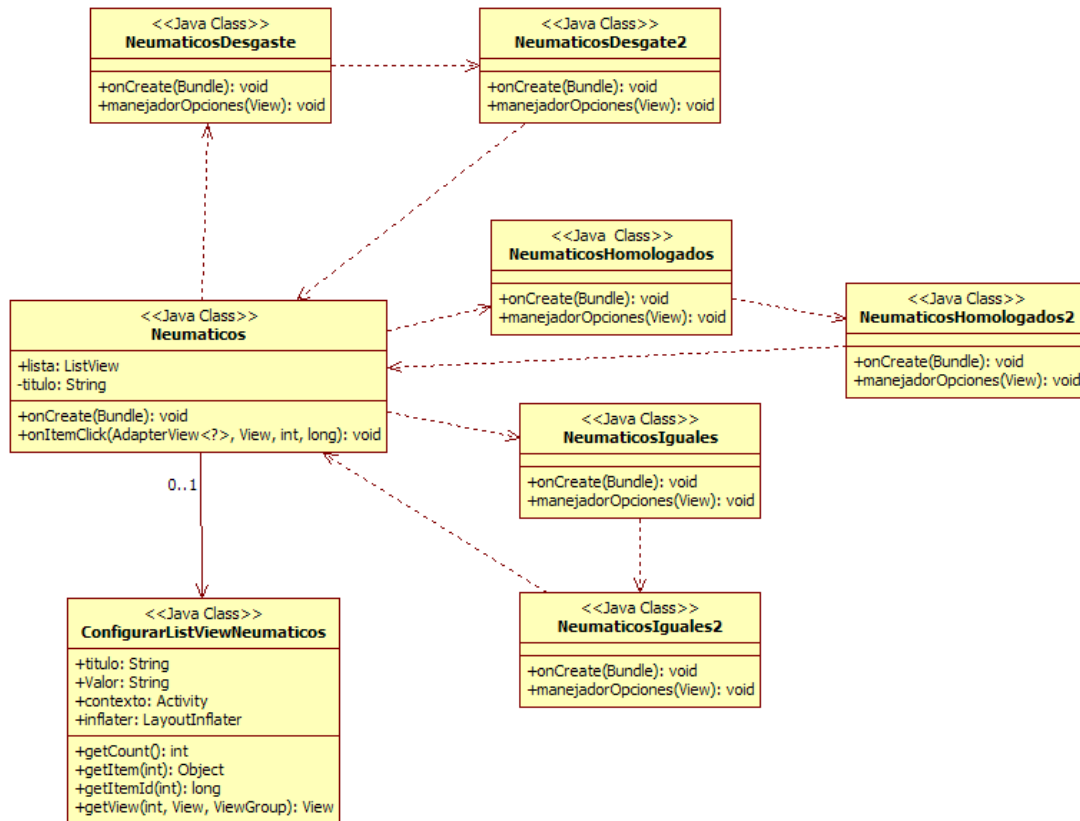


Figura 52: Diagrama de clases revisar neumáticos coche

En el siguiente diagrama Figura 53 es el que se corresponde a las clases para las revisiones en general del coche. Al igual que en los casos anteriores a éstas se accede desde la clase *General*. Una vez se accede a una de ellas, por ejemplo *GeneralAceite* se le proporciona en ella una información al usuario, y tiene la opción de acceder a la siguiente *GeneralAceite2* en la cual se le realizará una pregunta al usuario. Tras responde dicha pregunta, se volverá a actualizar la lista dependiendo de la respuesta dada por dicho usuario.

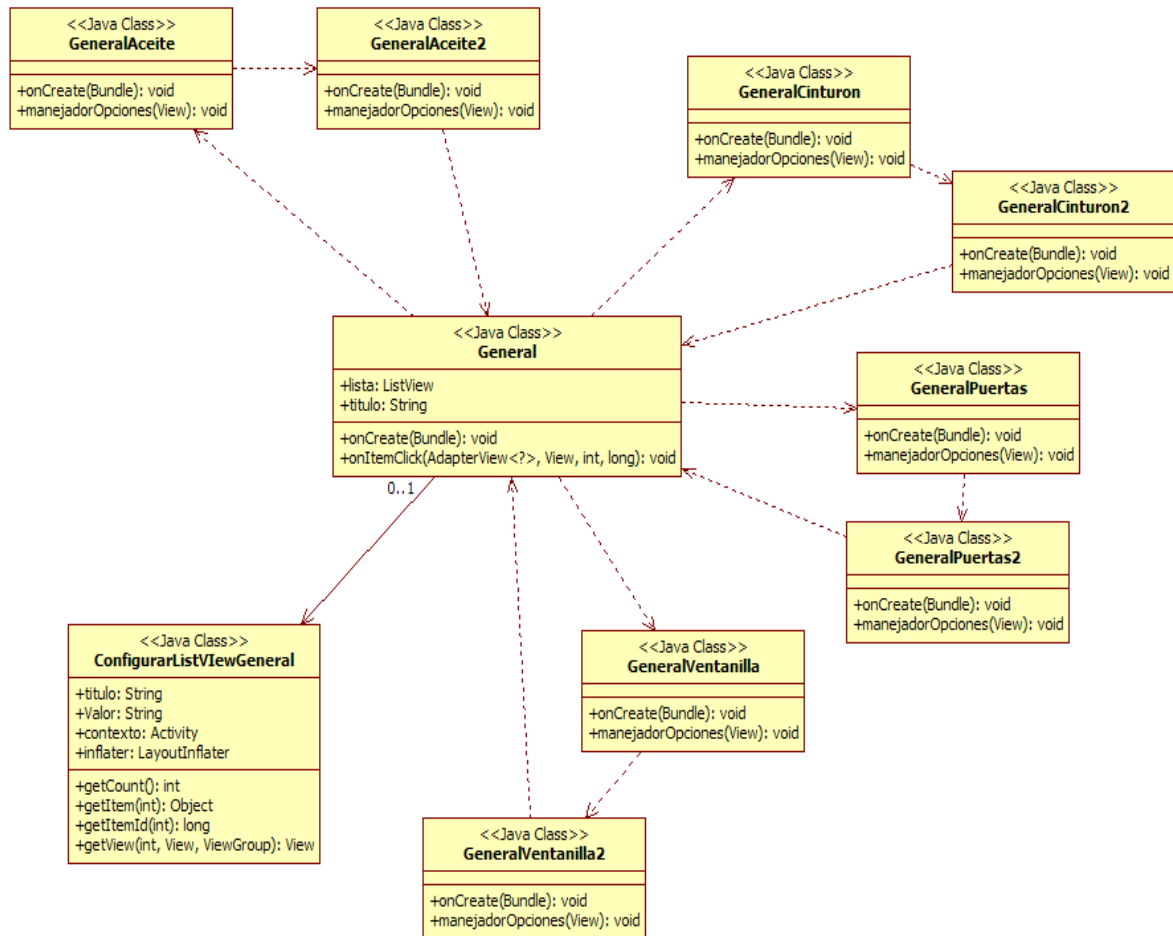


Figura 53: Diagrama de clases revisar general coche

Por otro lado, si la opción elegida para el tipo de vehículo seleccionado es la motocicleta las partes que podrán ser revisadas serán las que se muestran en los siguientes diagramas de clases.

El diagrama siguiente Figura 54, muestras las clases necesarias para mostrar las partes a revisar de las motocicletas. Como se puede observar son prácticamente idénticas a las vistas en el diagrama anterior, ya que su funcionalidad es la misma, puesto que realiza las mismas acciones que en el caso de los coches, solo que en este caso para motocicletas.

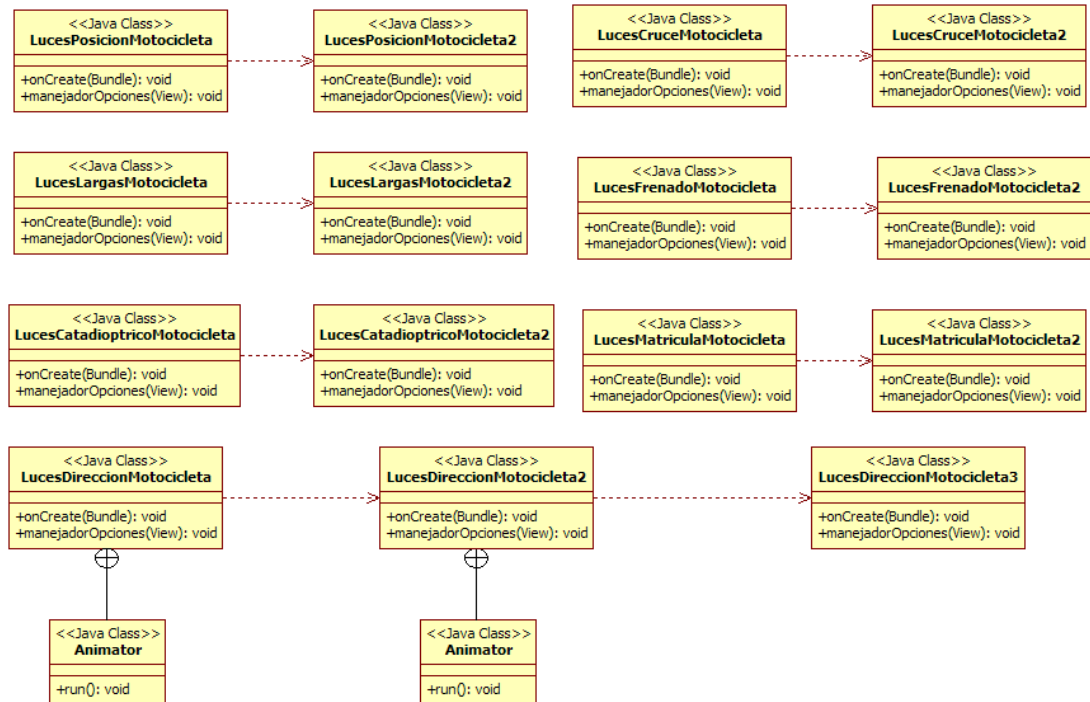


Figura 54: Diagrama de clases revisar alumbrado motocicleta

El siguiente diagrama, Figura 55, es el que se corresponde a la revisión de las partes del frenado de la motocicleta.

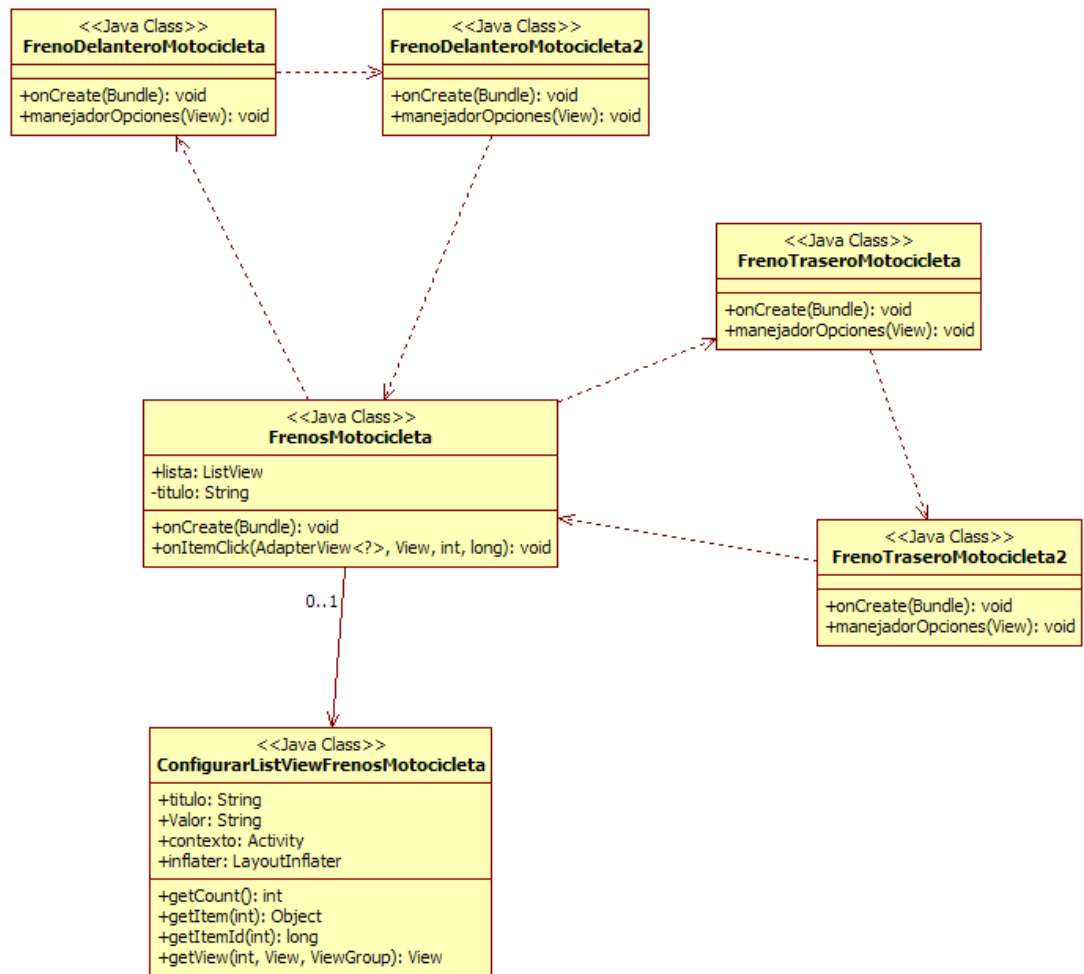


Figura 55: Diagrama de clases revisar frenos motocicleta

El diagrama que se puede observar a continuación, Figura 56, muestra las clases correspondientes a la revisión de los neumáticos de la motocicleta y son muy similares a los vistos anteriormente.

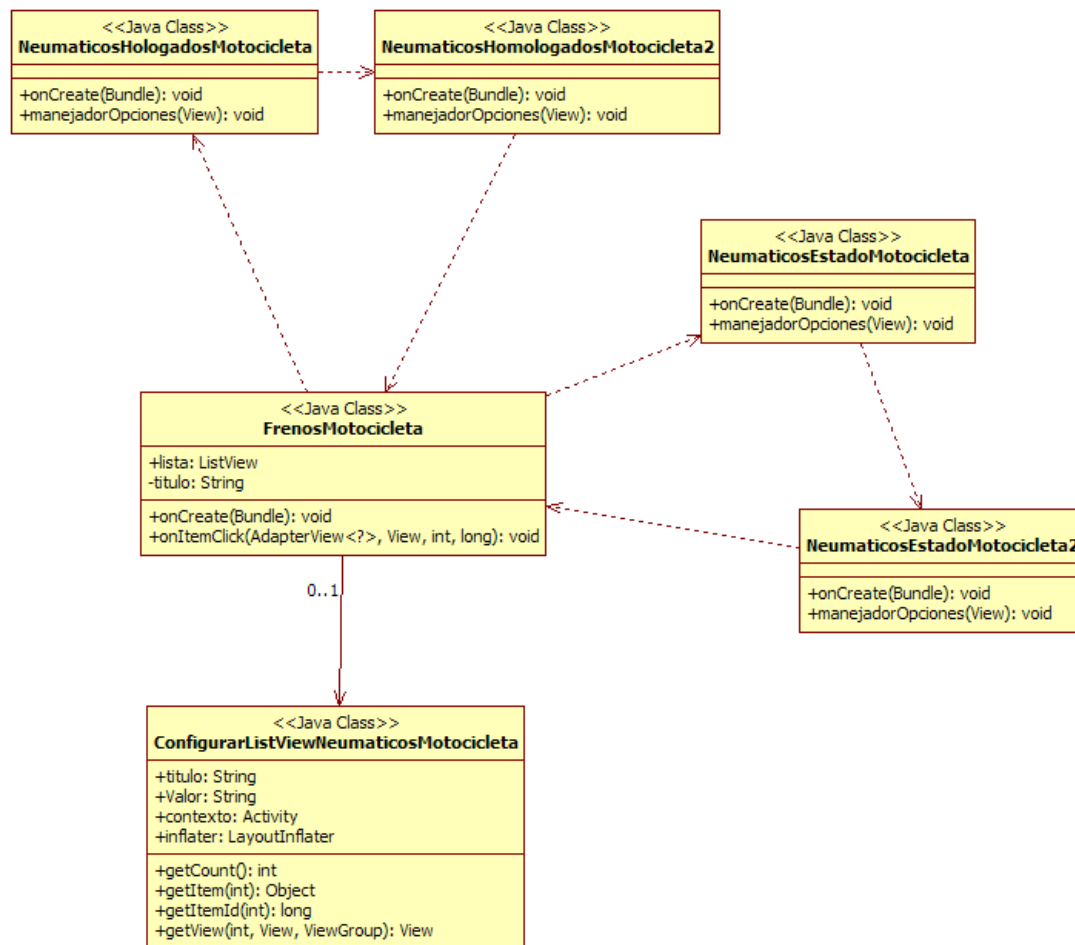


Figura 56: Diagrama de clases revisar neumáticos motocicleta

El siguiente diagrama es el que se corresponde a las clases de revisiones generales para lo opción de la motocicleta.

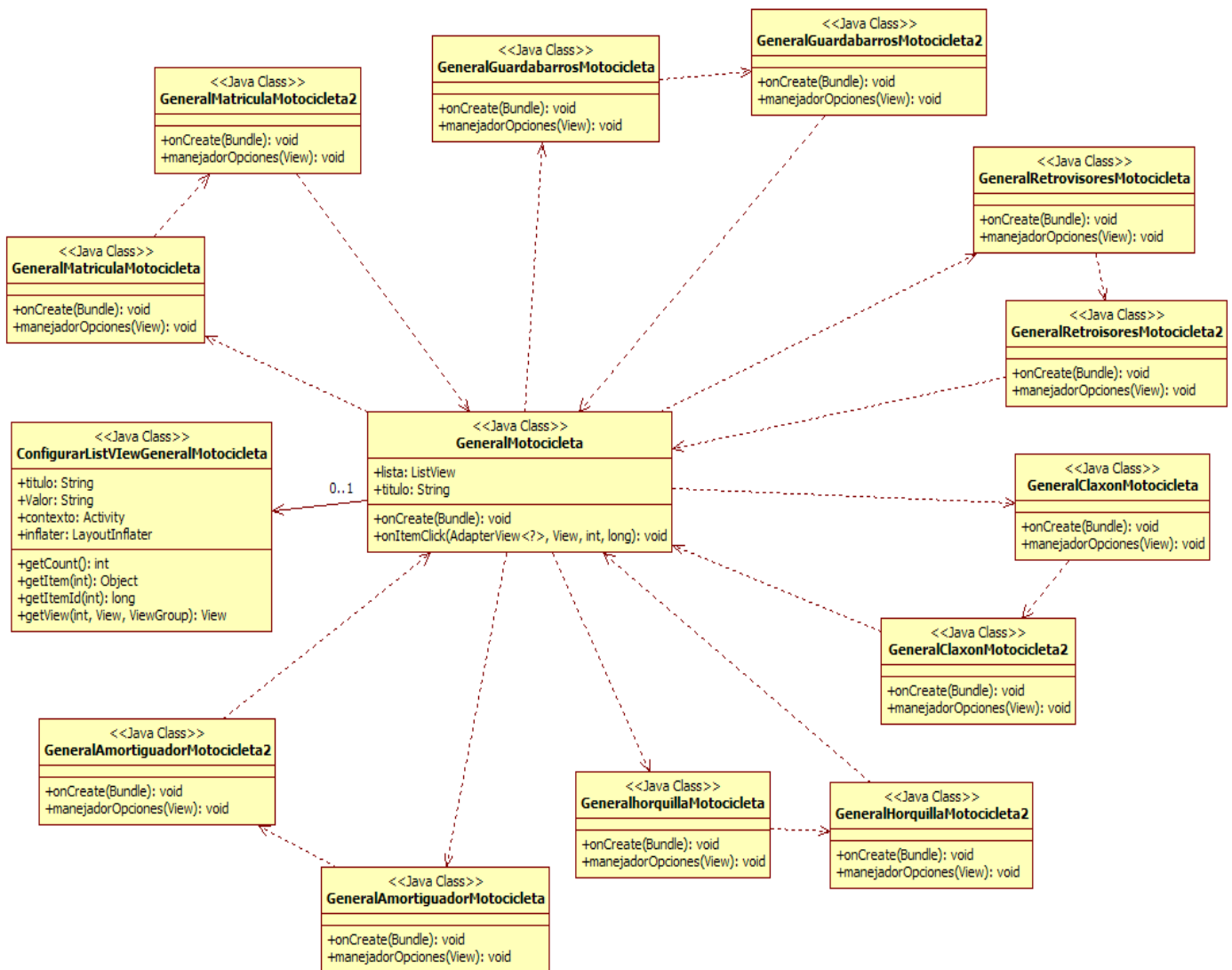


Figura 57: Diagrama de clases revisar general motocicleta

El siguiente diagrama Figura 58 es el relacionado con la funcionalidad de calcular la próxima revisión del vehículo.

Como se puede observar, se parte de la clase principal de la aplicación *MyIltvActivity*. A partir de esta clase se podrá acceder a la clase *MyIltvProximaRevision* en la cual se podrá calcular la próxima inspección introduciendo para ello la fecha de matriculación y el tipo de vehículo sobre el que se desee calcular. En esta misma clase, se tendrá la opción de poder ver los periodos con los que se realizan las inspecciones en cada tipo de vehículo, y, para ello será necesario acceder a la clase *InfoRevisiones*. Como se puede ver, en esta clase se encuentran los métodos necesarios para cargar

una lista expandible en la que se muestran todos los vehículos con sus respectivos periodos.

Por otro lado, desde la clase *MyITVProximaRevision* tambien se podrá acceder a la clase *ConfirmarRevision* en la cual se dará la opción al usuario para activar una notificación que le servirá para ser alertado de la próxima revisión. Una vez que el usuario seleccione la opción para activar la notificación se pasará a la siguiente clase *DialogoPersonalizado* mediante el cual el usuario podrá introducir tanto la matrícula de su vehículo, para que este quede identificado, como los días de antelacion con los que desea ser avisado. Una vez activada la notificación se pasará a la clase *AlarmReceiver*, que será la encargada de almacenar dicha notificación.

Finalmente, desde la clase principal *MyITVActivity* tambien se puede acceder a la clase *MisNotificaciones*, en las cuales se podrán visualizar todas las notificaciones que hayan sido guardadas con anterioridad mediante la clase *AlarmReceiver*, e incluso eliminarlas si se cree conveniente.

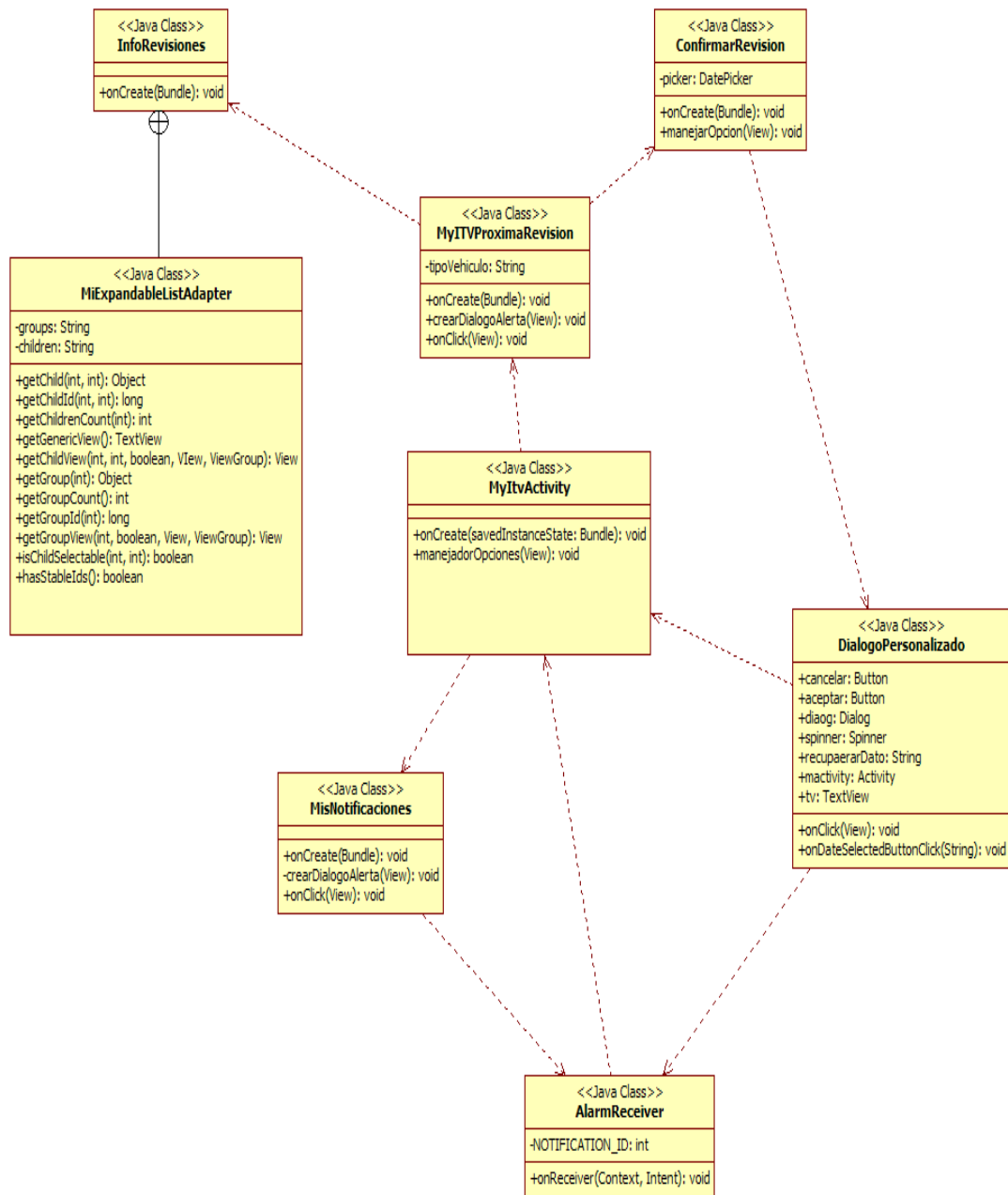


Figura 58: Diagrama de clases notificaciones y documentación

El siguiente diagrama, Figura 59 es el que se corresponde con la funcionalidad de la aplicación para ver la documentación necesaria para la inspección del vehículo.

Como se puede observar se parte de la actividad principal de la aplicación *MyItvActivity* nuevamente. Desde esta clase se podrá acceder a la clase *DocumentacionPermisoCirculacion* desde la cual se podrá acceder a la clase *DocumentacionFichaTecnica* y desde ésta última a la clase *DocumentacionSeguro*. En todas ellas se mostrará información sobre toda la documentación que será necesario presentar en la inspección para poder pasarla satisfactoriamente.

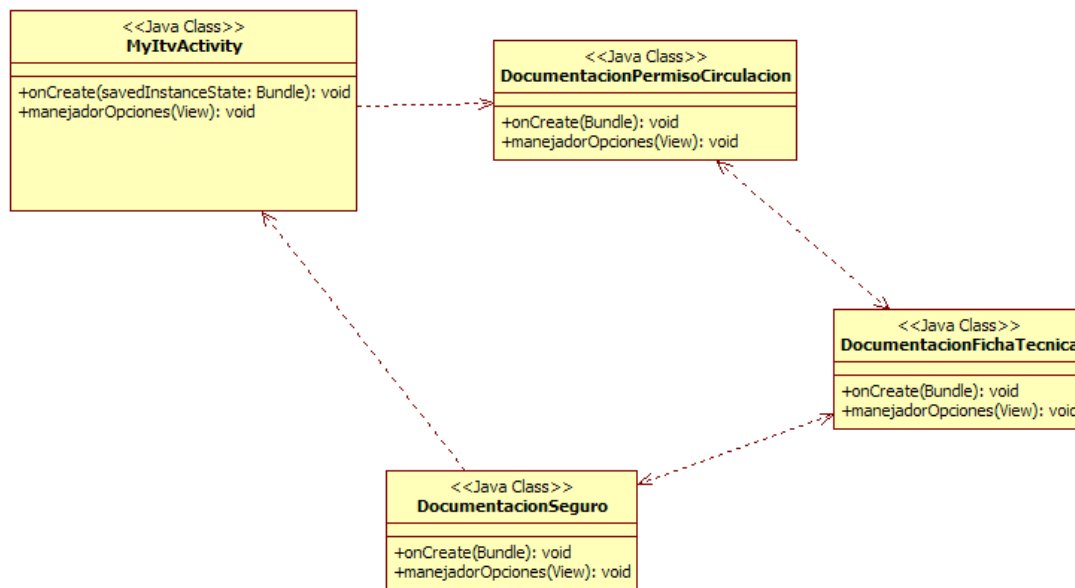


Figura 59: Diagrama de clases documentación

El último diagrama, Figura 60 es el relacionado con las clases utilizadas para todo el manejo de la base de datos.

Estas clases serán las encargadas de la creación de la base de datos y de gestionarla mediante sus métodos.

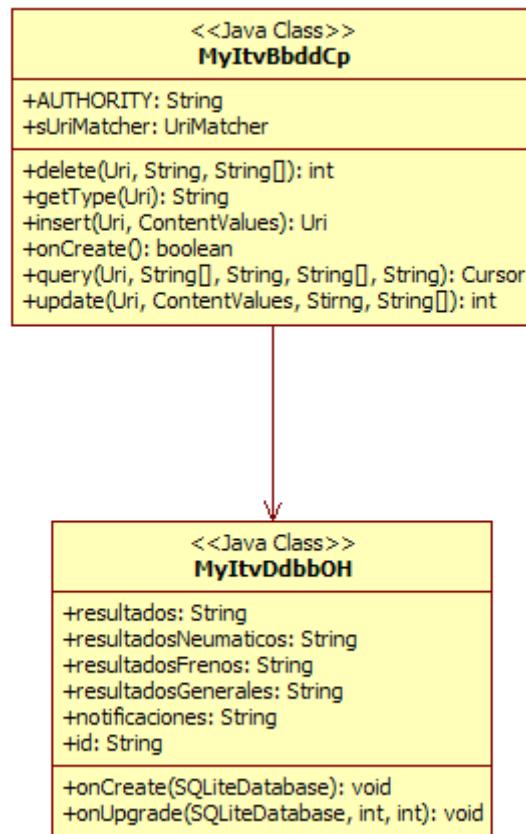


Figura 60: Diagrama de clases base de datos

A continuación se explicará en detalle todo lo relacionado con el manejo y diseño de la base de datos.

Durante el diseño de la aplicación se ha considerado necesario la inserción de una base de datos en la cual se almacenarán los diferentes elementos que se considere necesario.

Para llevar a cabo la inserción de dicha base de datos se ha hecho uso de los *content providers* [4], los cuales servirán para almacenar y recuperar datos y además éstos podrán ser accesibles a otras aplicaciones por si en líneas futuras esto fuera necesario.

Android viene equipado con un número determinado de *content provider*, los cuales se pueden ver en el paquete `android.provider`, pero para este caso crearemos uno propio.

Los *content providers* exponen sus datos en tablas sencillas de un modelo de base de datos, donde cada fila es un registro y cada columna es un dato de tipo y significado particular. Cada registro incluye un campo entero `_ID` que identifica unívocamente cada registro dentro de la tabla.

Cada consulta a la base de datos devuelve un objeto *Cursor* que avanza de un registro a otro y de columna en columna para leer los contenidos de cada campo.

Cada *content provider* expone un *URI* público que identifica unívocamente su conjunto de datos. Todas las *URIs* de los providers empiezan por el string `"content://"`, lo cual significa que los datos están controlados por un *content provider*.

En nuestro caso la *URI* construida para realizar las consultas a la base de datos será de la siguiente forma:

`content://com.tfg.myitv.bbdd.MyItvBbddCP/`

Ésta irá seguida del nombre de la tabla sobre la que se desea operar, por ejemplo si se tratara de la tabla "notificaciones", la *URI* quedaría de la siguiente forma:

`content://com.tfg.myitv.bbdd.MyItvBbddCP/notificaciones`

Para la creación de nuestro propio *content provider* será necesario definir una subclase *ContentProvider* en la cual se implementarán los métodos abstractos necesarios para su funcionamiento. A continuación se detalla los más relevantes:

- Método `onCreate()`: Su función es instanciar el objeto de la clase que se explicará más adelante y que creará las tablas y la base de datos. Este método es el que se puede observar en la Figura 61.

```
@Override
public boolean onCreate() {
    miDdbbHelper = new MyItvBbddOH(getContext());
    return true;
}
```

Figura 61: Creación de la base de datos

- Método `insert()`: encargado de insertar la información en la tabla correspondiente tal y como se puede observar en la Figura 62.

```
@Override
public Uri insert(Uri uri, ContentValues values) {
    SQLiteDatabase db = miDdbbHelper.getWritableDatabase();
    long rowId;
    switch (sUriMatcher.match(uri)) {
        case 1:
            rowId = db.insert("resultados", null, values);
            if(rowId>0)
                getContext().getContentResolver().notifyChange(uri, null);
            break;
        case 2:
            rowId = db.insert("notificaciones", null, values);
            if(rowId>0)
                getContext().getContentResolver().notifyChange(uri, null);
            break;
        case 3:
            rowId = db.insert("resultadosNeumaticos", null, values);
            if(rowId>0)
                getContext().getContentResolver().notifyChange(uri, null);
            break;
        case 4:
            rowId = db.insert("resultadosFrenos", null, values);
            if(rowId>0)
                getContext().getContentResolver().notifyChange(uri, null);
            break;
        case 5:
            rowId = db.insert("resultadosGenerales", null, values);
            if(rowId>0)
                getContext().getContentResolver().notifyChange(uri, null);
            break;
    }
    return null;
}
```

Figura 62: Método insert

- Método query(): Este método se encargará de devolvernos un *Cursor* con los datos deseados. Para ello usa la *URI* que recibe y busca en la tabla que se corresponda. La implementación de dicho método se puede observar en la Figura 63.

```

@Override
public Cursor query(Uri uri, String[] projection, String selection,String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();
    SQLiteDatabase db = miDdbbHelper.getReadableDatabase();
    Cursor results=null;
    switch (sUriMatcher.match(uri)) {
        case 1:
            queryBuilder.setTables("resultados");
            results = queryBuilder.query(db, projection, selection,selectionArgs, null, null, sortOrder);
            break;

        case 2:
            queryBuilder.setTables("notificaciones");
            results = queryBuilder.query(db, projection, selection,selectionArgs, null, null, sortOrder);
            break;

        case 3:
            queryBuilder.setTables("resultadosNeumaticos");
            results = queryBuilder.query(db, projection, selection,selectionArgs, null, null, sortOrder);
            break;

        case 4:
            queryBuilder.setTables("resultadosFrenos");
            results = queryBuilder.query(db, projection, selection,selectionArgs, null, null, sortOrder);
            break;

        case 5:
            queryBuilder.setTables("resultadosGenerales");
            results = queryBuilder.query(db, projection, selection,selectionArgs, null, null, sortOrder);
            break;

        default:
            throw new IllegalArgumentException("Invalid URI " + uri);
    }

    //results.setNotificationUri(getContext().getContentResolver(), uri);
    return results;
}

```

Figura 63: Método query

- Método delete(): Con este método se eliminarán los datos que se deseen de la tabla elegida. La implementación de dicho método se puede observar en la Figura 64.

```
@Override
public int delete(Uri uri, String selection, String[] selectionArgs) {
    SQLiteDatabase db = miDdbbHelper.getWritableDatabase();
    int count;
    switch (sUriMatcher.match(uri)) {
        case 1:
            count = db.delete("resultados", selection, null);
            break;
        case 2:
            count = db.delete("notificaciones", selection, null);
            break;
        case 3:
            count = db.delete("resultadosNeumaticos", selection, null);
            break;
        case 4:
            count = db.delete("resultadosFrenos", selection, null);
            break;
        case 5:
            count = db.delete("resultadosGenerales", selection, null);
            break;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
```

Figura 64: Método delete

- Método update(): Con este método se podrá actualizar un registro de una determinada tabla de la base de datos. La implementación de dicho método es la que se puede observar en la Figura 65.

```

@Override
public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
    int count = 0;

    SQLiteDatabase sqlDB = miDdbbHelper.getWritableDatabase();

    switch (sUriMatcher.match(uri)) {
        case 1:
            count = sqlDB.update("resultados", values, selection, selectionArgs);
            break;
        case 2:
            count = sqlDB.update("notificaciones", values, selection, selectionArgs);
            break;
        case 3:
            count = sqlDB.update("resultadosNeumaticos", values, selection, selectionArgs);
            break;
        case 4:
            count = sqlDB.update("resultadosFrenos", values, selection, selectionArgs);
            break;
        case 5:
            count = sqlDB.update("resultadosGenerales", values, selection, selectionArgs);
            break;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}

```

Figura 65: Método update

Una vez creados estos métodos también se deberá crear en otra nueva clase que extienda de la clase “SQLiteOpenHelper” y en la cual se generarán las tablas que se crea necesario. En este caso se han creado cinco tablas, cuatro de ellas para almacenar los resultados que se van generando tras ir realizando las preguntas en la parte de “¿Qué Revisar?”, y una última para ir almacenando las fechas de las notificaciones.

La estructura de las cuatro primeras tablas es la que se puede observar en la Figura 66.

```

String resultados = "CREATE TABLE resultados (_id INTEGER PRIMARY KEY autoincrement, Posicion TEXT, Cruce TEXT, Carretera TEXT, Marcha_atrás TEXT, Dirección TEXT, " +
    "Emergencia TEXT, Frenado TEXT, Matricula TEXT, Antiniebla_Del TEXT, Antiniebla_Tras TEXT)";
String resultadosNeumaticos = "CREATE TABLE resultadosNeumaticos (_id INTEGER PRIMARY KEY autoincrement, Homologacion TEXT, Desgaste TEXT, Mismos TEXT)";
String resultadosFrenos = "CREATE TABLE resultadosFrenos (_id INTEGER PRIMARY KEY autoincrement, Vibracion TEXT, Freno_Fuerte TEXT, Ruido TEXT, Freno_Mano)";
String resultadosGenerales = "CREATE TABLE resultadosGenerales (_id INTEGER PRIMARY KEY autoincrement, Aceite TEXT, Cinturones TEXT, Ventanillas TEXT, Puertas)";

```

Figura 66: Estructura base de datos

Estas cuatro primeras tablas solo van a tener una fila con todos sus campos, los cuales se irán actualizando de manera que se vayan respondiendo negativa o positivamente las preguntas de la parte “¿Qué Revisar?”.

Por otro lado, la tabla “notificaciones” contendrá más de una sola fila, ya que podrá existir más de una notificación, pero solo podrá haber una por matrícula de vehículo. La estructura de dicha tabla será la que se puede observar en la Figura 67.

```
String notificaciones = "CREATE TABLE notificaciones (_id INTEGER PRIMARY KEY autoincrement,Matricula TEXT,Fecha TEXT, Identificador TEXT)";
```

Figura 67: Estructura base de datos 2

Después de esto solo quedará ejecutar el método `execSQL()` para que se generen las tablas tal y como se puede apreciar en la Figura 68.

```
@Override
public void onCreate(SQLiteDatabase database) {

    database.execSQL(resultados);
    database.execSQL(resultadosNeumaticos);
    database.execSQL(resultadosGenerales);
    database.execSQL(resultadosFrenos);
    database.execSQL(notificaciones);

}
```

Figura 68: Ejecución base de datos

Cabe destacar que las bases de datos de Android se almacenan dentro del propio dispositivo, y para poder borrarlas será necesario desinstalar la aplicación.

4.2.5 MODELO ENTIDAD-RELACIÓN BASE DE DATOS

A continuación se mostrará el modelo entidad-relación de los datos, en el cual se representarán las entidades relevantes del sistema de información así como sus interrelaciones y propiedades. Dicho modelo es el que se puede observar en la Figura 69.

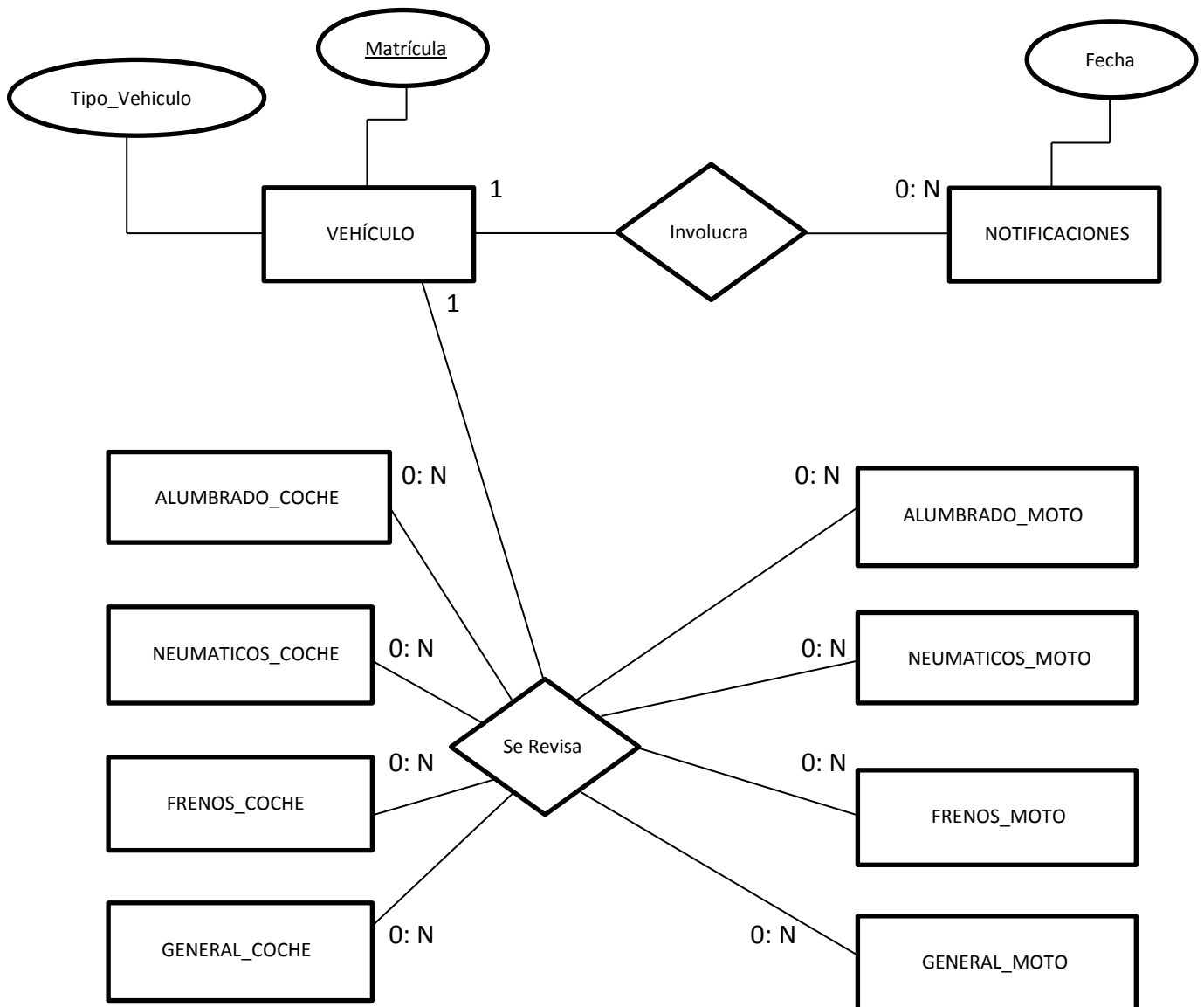


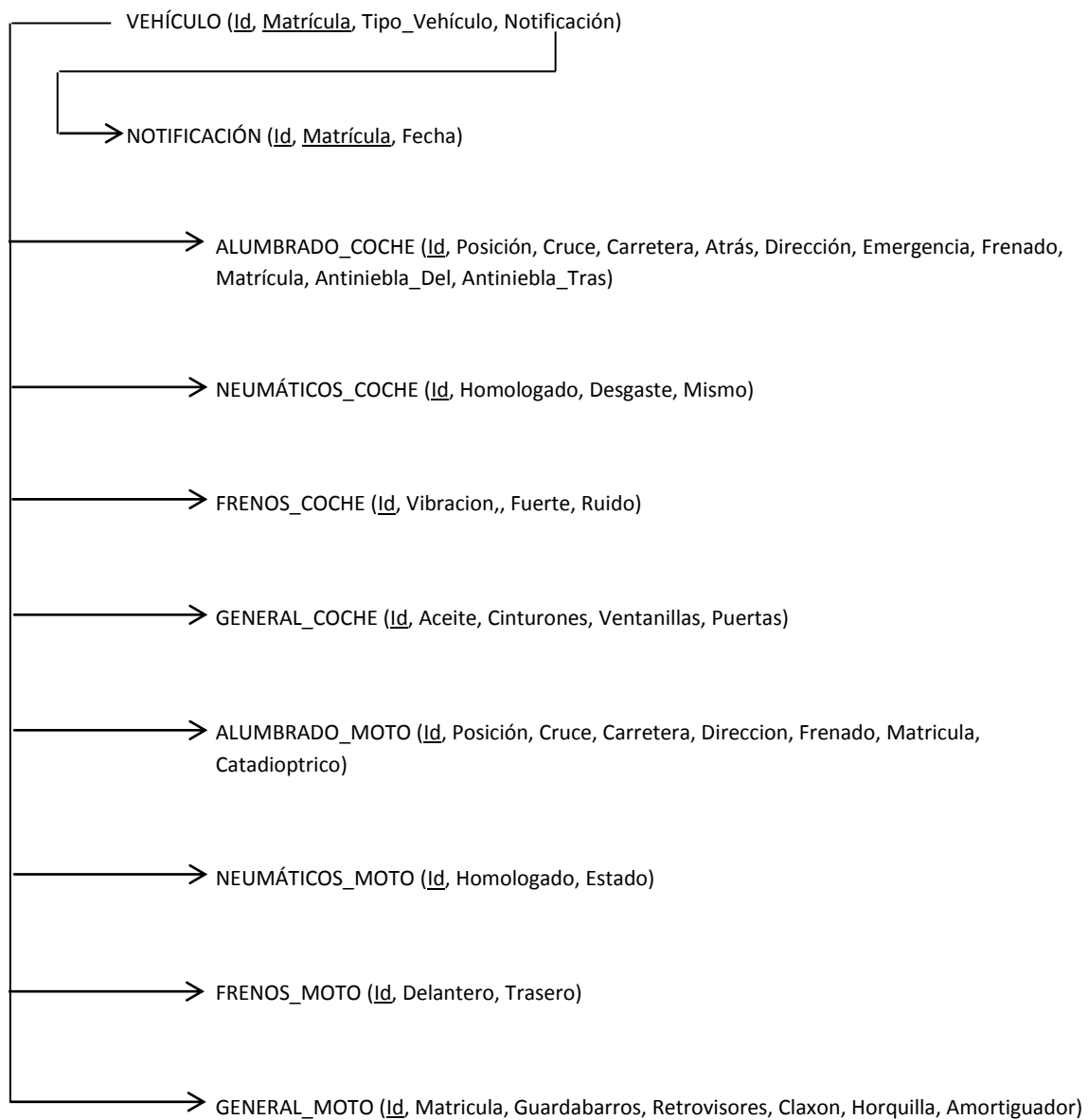
Figura 69: Modelo entidad-relación de los datos

Como se puede observar en la figura anterior, las composiciones para las partes del vehículo que se pueden revisar no han sido completadas con sus atributos, puesto que el esquema quedaría demasiado grande y no se apreciaría con demasiada claridad, por lo que se ha decidido obviar.

4.2.6 DISEÑO DEL MODELO FÍSICO DE DATOS

En este apartado se definirá la estructura física de datos que utilizará el sistema, de manera que se consiga una mayor eficiencia en el tratamiento de los datos.

A continuación se puede observar el modelo físico de datos utilizado, con las respectivas tablas y campos y como se relacionan entre ellas para conseguir un adecuado tratamiento de la información.



5 PRUEBAS DE LA APLICACIÓN

5	PRUEBAS DE LA APLICACIÓN	131
5.1	PRUEBAS DE ACEPTACIÓN	132

En este capítulo se van a especificar las pruebas necesarias para garantizar el correcto funcionamiento de la aplicación. Todas las pruebas especificadas deberán ser superadas con éxito para considerar que el proyecto cumple con las expectativas esperadas.

5.1 PRUEBAS DE ACEPTACIÓN

A continuación se detallarán todas las pruebas de aceptación del sistema que se han llevado a cabo para comprobar el correcto funcionamiento del mismo. Las pruebas serán representadas en tablas como la que se puede observar en Tabla 37.

IDENTIFICADOR	PA – XX
Descripción	
Entrada esperada	
Salida esperada	
Errores posibles	
Requisitos	
Estado	

Tabla 37: Formato tabla pruebas de aceptación

Los campos de las pruebas serán los siguientes

- **Identificador:** será el código que identificará a cada prueba. Dicho código será único para cada una de las diferentes pruebas. Estará compuesto por dos partes separadas por un guion. La primera parte serán las siglas de Prueba de Aceptación (PA) seguido del número que identificará a cada una de ellas.
- **Descripción:** objetivo que persigue la prueba.
- **Entrada esperada:** acción de entrada al sistema que provocará una actuación del mismo.
- **Salida esperada:** resultado que devolverá el sistema tras la entrada correspondiente.
- **Errores posibles:** posibles errores que podrían surgir tras realizar la prueba, o que se han podido ir encontrando y que por lo tanto no permitirían que la prueba sea válida.
- **Requisitos:** se nombrarán los requisitos con los que guardan relación cada una de las pruebas.

- **Estado:** muestra como ha finalizado la prueba. Si el resultado es correcto se marcará como “correcta” y en el caso contrario se marcará como “incorrecta”.

A continuación se mostrará cada una de las pruebas llevadas a cabo con sus respectivos resultados.

IDENTIFICADOR	PA – 01
Descripción	Inicio correcto de la aplicación.
Entrada esperada	Pulsar sobre el icono de la aplicación y lanzarla.
Salida esperada	La aplicación debería iniciarse y mostrar las cuatro opciones disponibles al usuario.
Errores posibles	La aplicación lanza un mensaje de error y no carga la aplicación.
Requisitos	N/A
Estado	CORRECTO.

Tabla 38: PA - 01 - Inicio correcto aplicación

IDENTIFICADOR	PA – 02
Descripción	Carga lista de alumbrado.
Entrada esperada	Pulsar sobre la opción de alumbrado.
Salida esperada	Comprobar que la lista con las partes del alumbrado carga correctamente con todos sus elementos.
Errores posibles	La lista no se carga correctamente. La lista no se actualiza de forma correcta tras realizar modificaciones en ella. Falta algún elemento de la lista. Al pulsar sobre algún elemento de la lista la aplicación lanza un error.
Requisitos	RF – 005
Estado	CORRECTO.

Tabla 39: PA - 02 - Cargar lista alumbrado

IDENTIFICADOR	PA – 03
Descripción	Carga lista de frenos.
Entrada esperada	Pulsar sobre la opción de los frenos.
Salida esperada	Comprobar que la lista con las partes de los frenos carga correctamente con todos sus elementos.
Errores posibles	La lista no se carga correctamente. La lista no se actualiza de forma correcta tras realizar modificaciones en ella. Falta algún elemento de la lista. Al pulsar sobre algún elemento de la lista la aplicación lanza un error.
Requisitos	RF – 007
Estado	CORRECTO.

Tabla 40: PA - 04 - Cargar lista frenos

IDENTIFICADOR	PA – 04
Descripción	Carga lista de los neumáticos.
Entrada esperada	Pulsar sobre la opción de comprobar neumáticos.
Salida esperada	Comprobar que la lista con las partes de los neumáticos carga correctamente con todos sus elementos.
Errores posibles	La lista no se carga correctamente. La lista no se actualiza de forma correcta tras realizar modificaciones en ella. Falta algún elemento de la lista. Al pulsar sobre algún elemento de la lista la aplicación lanza un error.
Requisitos	RF – 006
Estado	CORRECTO.

Tabla 41: PA - 04 - Cargar lista neumáticos

IDENTIFICADOR	PA – 05
Descripción	Carga lista de preparativos generales.
Entrada esperada	Pulsar sobre la opción de preparativos generales.
Salida esperada	Comprobar que la lista con las partes generales carga correctamente con todos sus elementos.
Errores posibles	La lista no se carga correctamente. La lista no se actualiza de forma correcta tras realizar modificaciones en ella. Falta algún elemento de la lista. Al pulsar sobre algún elemento de la lista la aplicación lanza un error.
Requisitos	RF – 008
Estado	CORRECTO.

Tabla 42: Cargar lista generales

IDENTIFICADOR	PA – 06
Descripción	Botón “home” correcto.
Entrada esperada	Pulsar en todas las pantallas sobre el botón de “home”.
Salida esperada	Comprobar que en todos los casos se accede a la pantalla inicial de la aplicación.
Errores posibles	La pantalla a la que se accede tras pulsar el botón no es la inicial. Tras pulsar el botón la pantalla lanza una excepción. Al pulsar el botón la aplicación se queda en la misma pantalla sin realizar ninguna acción.
Requisitos	RF – 016
Estado	CORRECTO.

Tabla 43: PA - 06 - Botón "home"

IDENTIFICADOR	PA – 07
Descripción	Lista de información de periodos.
Entrada esperada	Se pulsa sobre el botón de información para periodos de revisión de cada vehículo.
Salida esperada	Debe aparecer una lista en la que se muestra el periodo para la inspección de cada tipo de vehículo.
Errores posibles	Tras pulsar el botón la aplicación permanece en el mismo estado. La lista no se carga correctamente o faltan elementos. Tras pulsar el botón la aplicación lanza un error.
Requisitos	RF – 011
Estado	CORRECTO.

Tabla 44: PA - 07 - Lista periodos

IDENTIFICADOR	PA – 08
Descripción	Cálculo correcto fecha inspección.
Entrada esperada	Se introduce el tipo de vehículo y la fecha de matriculación del mismo.
Salida esperada	Se debe devolver la fecha correcta de la próxima inspección del vehículo.
Errores posibles	No se muestra la fecha de la próxima inspección. La fecha que se muestra no es la correcta.
Requisitos	RF – 002, RF – 009, RF – 010
Estado	CORRECTO.

Tabla 45: PA - 08 - Cálculo fecha inspección

IDENTIFICADOR	PA – 09
Descripción	Activación de notificaciones.
Entrada esperada	Se introduce la matrícula del vehículo y la fecha en la que se quiere ser avisado.
Salida esperada	El usuario debe ser avisado mediante una notificación en la fecha seleccionada de que tiene una inspección pendiente.
Errores posibles	La notificación no aparece. La notificación aparece en una fecha diferente a la deseada por el usuario.
Requisitos	RF – 012, RF – 013, RF – 014
Estado	CORRECTO.

Tabla 46: PA - 09 - Activar notificación

IDENTIFICADOR	PA – 10
Descripción	Mostrar notificaciones guardadas.
Entrada esperada	Se introduce la matrícula del vehículo y la fecha en la que se quiere ser avisado.
Salida esperada	Se mostrarán todas las notificaciones guardadas con las fechas y matrículas correspondientes.
Errores posibles	No aparecen las notificaciones guardadas. La matrícula del vehículo no aparece correctamente. La fecha que aparece no es la correcta.
Requisitos	RF – 012, RF – 003
Estado	CORRECTO.

Tabla 47: PA - 10 - Mostrar notificaciones

IDENTIFICADOR	PA – 11
Descripción	Eliminar notificaciones.
Entrada esperada	Se pulsa sobre el botón de eliminar notificaciones.
Salida esperada	La notificación que se desee eliminar quedará borrada.
Errores posibles	No se elimina la notificación deseada y sigue apareciendo en la pantalla destinada a ello. A pesar de ser eliminada la notificación sigue apareciendo en la fecha en la que se seleccionó.
Requisitos	RF – 012, RF – 015
Estado	CORRECTO.

Tabla 48: PA - 11 - Eliminar notificaciones

IDENTIFICADOR	PA – 12
Descripción	Transición correcta entre todas las pantallas de la aplicación.
Entrada esperada	Se accede desde una pantalla a otra pasando por todas las de la aplicación.
Salida esperada	Se avanzará de una pantalla a otra de la aplicación correctamente.
Errores posibles	No se avanza a la pantalla correcta. La aplicación lanza un error al intentar avanzar a una pantalla.
Requisitos	N/A
Estado	CORRECTO.

Tabla 49: PA - 12 - Transición correcta actividades

IDENTIFICADOR	PA – 12
Descripción	Imágenes correctas en la aplicación.
Entrada esperada	Se accede a todas las pantallas de la aplicación.
Salida esperada	Se deberían cargar correctamente todas las imágenes de la aplicación.
Errores posibles	No aparece la imagen. La imagen no aparece en la posición que debería aparecer.
Requisitos	N/A
Estado	CORRECTO.

Tabla 50: Imágenes correctas aplicación

6 GESTION DEL PROYECTO

6	GESTION DEL PROYECTO	140
6.1	PLANIFICACIÓN DEL PROYECTO	141
6.1.1	PLANIFICACIÓN INICIAL	141
6.1.2	PLANIFICACIÓN FINAL.....	144
6.2	HERRAMIENTAS UTILIZADAS	146
6.2.1	HARDWARE.....	146
6.2.2	SOFTWARE.....	147
6.3	ANÁLISIS ECONÓMICO	147

Este capítulo servirá para explicar los aspectos relacionados con la gestión del proyecto. Para ello inicialmente se procederá a explicar todas las fases que se han llevado a cabo en el proyecto. Seguidamente se especificarán todas las herramientas, tanto de software como hardware que se han requerido para el desarrollo de la misma. Finalmente se abordará el presupuesto que supondrá el desarrollo de la aplicación.

6.1 PLANIFICACIÓN DEL PROYECTO

En este apartado se explicarán todas las tareas que se han llevado a cabo a lo largo del desarrollo del proyecto. Para ello se hará uso de un diagrama de Gantt en el cual se podrá ver con exactitud los periodos de tiempo que se han necesitado para cada una de dichas tareas. Además, se realizará una comparación entre la planificación que se realizó inicialmente y la que finalmente se llevó a cabo.

6.1.1 PLANIFICACIÓN INICIAL

La primera tarea que se va a poder encontrar en el proyecto es la tarea de **planificación**. Esta tarea consistirá en realizar un cálculo de los días que serán necesarios para realizar cada una de las tareas hasta la fecha límite. Se considera como fecha límite el día 4 de Septiembre de 2012.

Seguidamente se iniciarían las tareas relacionadas con el **análisis** del proyecto. Esta tarea consiste en el estudio de otras aplicaciones similares a la pretendida y que por lo tanto pudieran aportar ideas o sugerencias para el desarrollo de la misma. Además también se analizará el entorno de desarrollo sobre el que se trabajará y la arquitectura elegida, acompañado de los casos de uso y los requisitos de usuario correspondientes.

A continuación y antes de que la fase de análisis finalizara, se iniciará la tarea de **diseño** de la aplicación. En esta fase se realizará un prototipo que servirá para tener una clara visión de como será la aplicación acompañado de los diagramas necesarios y del diseño de la base de datos utilizado.

Con la fase de diseño comenzada, dará inicio la fase de **implementación y codificación**. En esta fase se comenzará con el desarrollo de la aplicación mediante el entorno de desarrollo escogido.

Una vez finalizada la tarea anterior se dará paso a la fase de **Pruebas** con la cual se comprobará que todo funciona como debe y de forma correcta.

Además, y paralelamente a todas estas fases del proyecto, se irá realizando la presente documentación. Para ello se irá desarrollando dicha documentación a la vez que se desempeña cada una de las fases del proyecto.

El diagrama de Gantt resultante para todas las fases del proyecto es el que se puede observar en la Figura 70.

Trabajo Fin de Grado

MyItv - Asistente móvil para la realización de preparativos para la ITV de un automóvil

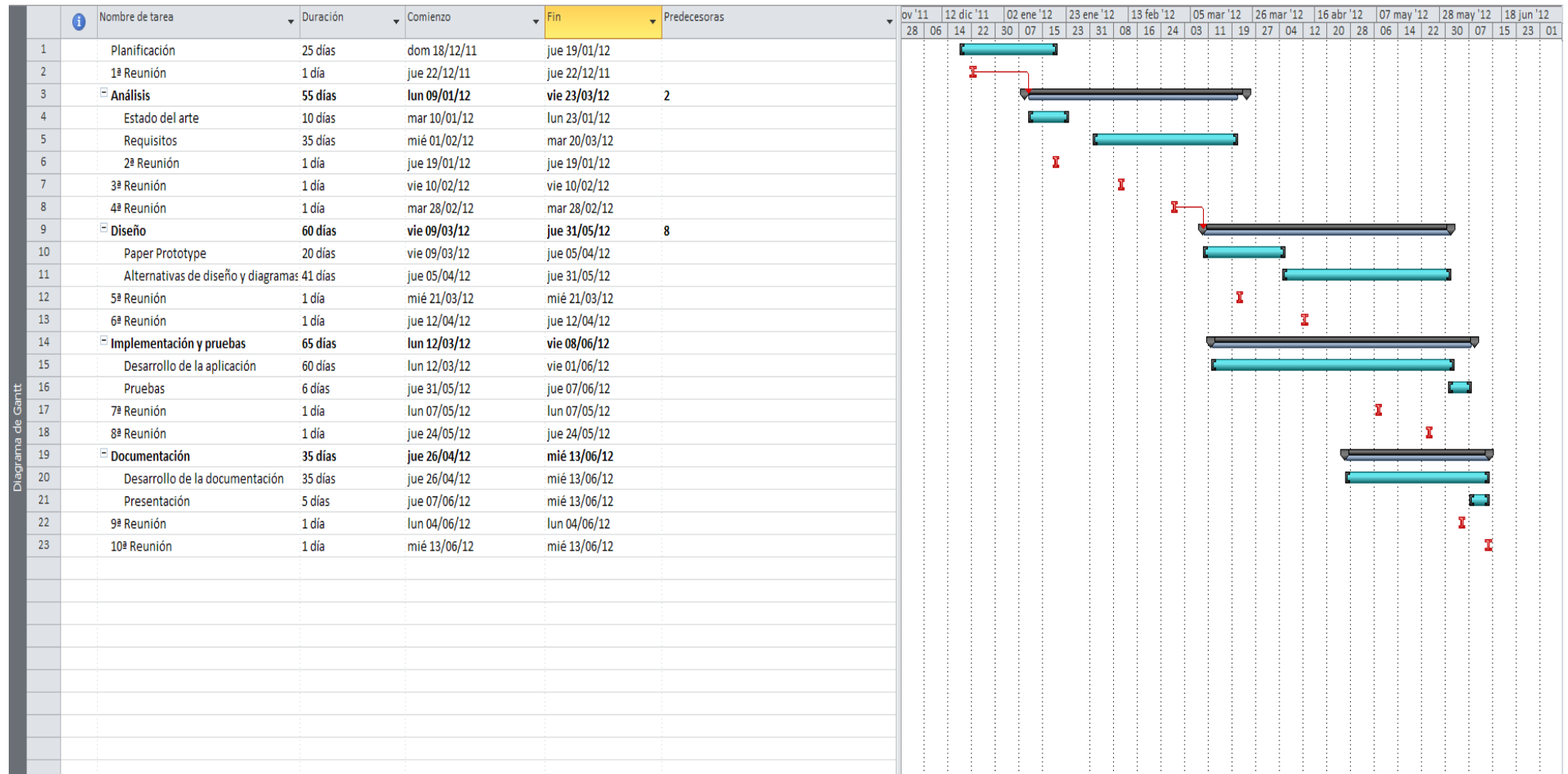


Figura 70: Diagrama de Gantt planificación inicial

Los puntos de color rojo que se pueden observar en el anterior diagrama pertenecen a las reuniones de seguimiento con el tutor. Como se puede observar se realizarán en periodos de unos 20 días mas o menos.

6.1.2 PLANIFICACIÓN FINAL

La planificación inicial se ha visto modificada a lo largo del desarrollo del proyecto. Estos cambios han sido bastante significativos, ya que, inicialmente se preveía que la finalización del mismo sería para mediados de Junio, y finalmente la fecha de finalización del mismo ha sido para comienzos de septiembre del mismo año.

El motivo por el que se ha sufrido este retraso ha sido principalmente la carga de tiempo que han supuesto las asignaturas cursadas durante el último cuatrimestre de la carrera y que por lo tanto hacían imposible compaginar ambas cosas. Es por ello que se decidió posponer la realización del presente proyecto para la fecha límite a día 5 de Septiembre de 2012.

El diagrama final que surge tras estos cambios es el que se puede observar en la Figura 71. Como se puede observar, todas las fases del desarrollo han sido respetadas, lo único que ha variado ha sido sus fechas de ejecución.

Trabajo Fin de Grado

MyItv - Asistente móvil para la realización de preparativos para la ITV de un automóvil

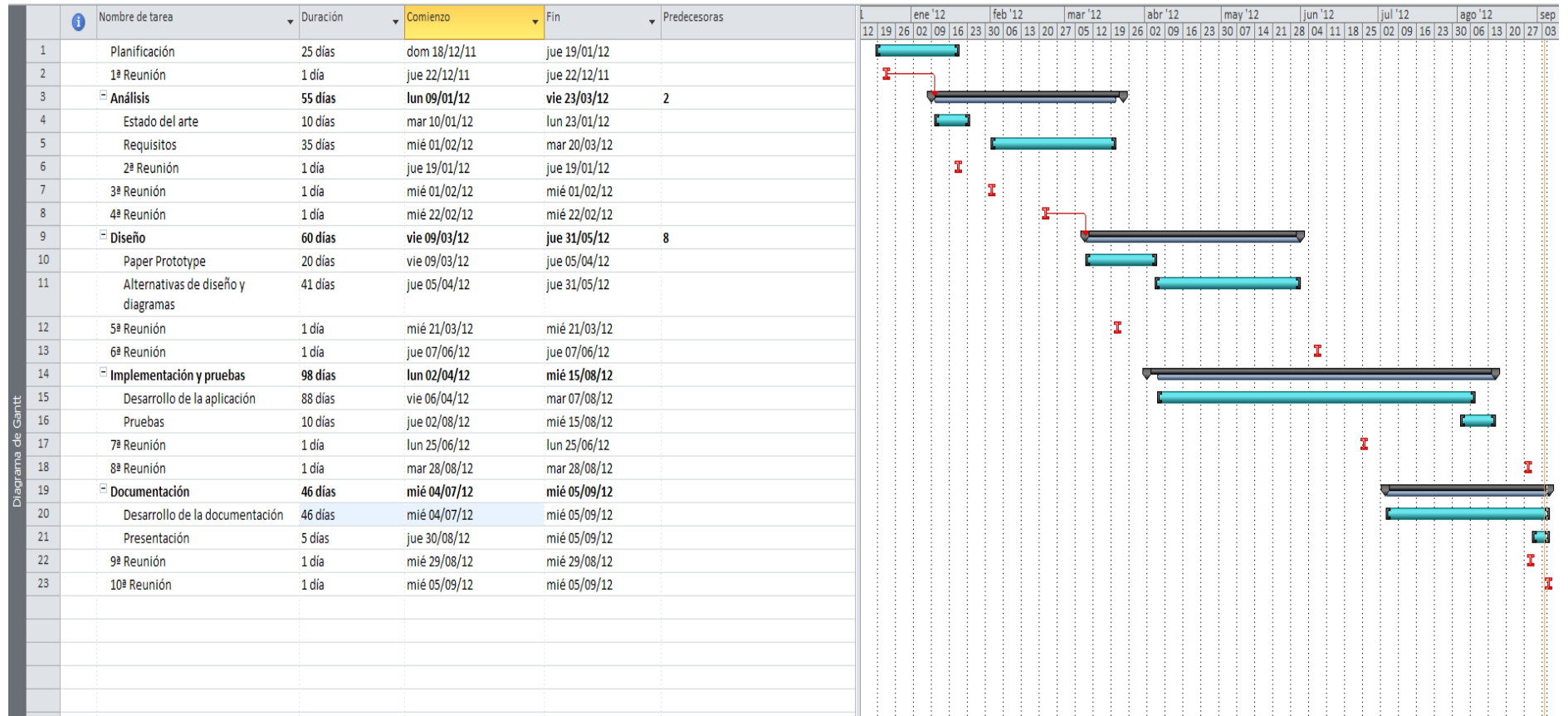


Figura 71: Diagrama de Gantt final

6.2 HERRAMIENTAS UTILIZADAS

En este apartado se detallarán las herramientas empleadas para el desarrollo de la aplicación. Se detallarán tanto las herramientas de tipo hardware como las herramientas de tipo software empleadas.

6.2.1 HARDWARE

Las herramientas de tipo hardware empleadas han sido por un lado un ordenador de sobremesa en el que se ha desarrollado toda la aplicación y por otro lado dos dispositivos móviles que han servido para comprobar los resultados de todo lo que se desarrollaba. A continuación se representa mediante unas tablas las características de cada uno de los dispositivos nombrados:

Dispositivo	Ordenador de Sobremesa
Procesador	Intel Pentium Dual Core a 2.40 GHz
Tarjeta Gráfica	NVIDIA GeForce 9400 GT
Memoria Ram	4,00 GB
Disco Duro	500 GB
Sistema Operativo	Windows 7 Ultimate

Tabla 51: Hardware ordenador sobremesa

Dispositivo	SmartPhone HTC Wildfire
Procesador	Procesador Qualcomm MSM 7225 a 528 MHz
Memoria Ram	384 MB RAM; 512 MB ROM memoria interna
Sistema Operativo	Android 2.3.7 (Actualizada)

Tabla 52: Hardware smartphone Wildfire

Dispositivo	SmartPhone Huawei Ascend G300
Procesador	Qualcomm MSM7227A Snapdragon S1 a 1 GHz
Memoria Ram	512 Mbytes de RAM
Sistema Operativo	Android 2.3.6

Tabla 53: Hardware smartphone Huawei

6.2.2 SOFTWARE

En este apartado se van a especificar las herramientas software que han sido utilizadas para el desarrollo de la aplicación. Estas herramientas se detallan en la siguiente tabla:

Nombre Herramienta	Tipo de Heramienta
Eclipse Indigo Release (3.7)	Entorno de Desarrollo Integrado
Editor de Imágenes	Adobe Photoshop 6.0
Diseño de diagramas	StarUML
Procesador de Textos y conversor a PDF	Microsoft Office Word 2010
Diseño de diagramas de Gantt	Microsoft Office Project 2010

Tabla 54: Software utilizado

6.3 ANÁLISIS ECONÓMICO

En este apartado se especificarán todos los detalles relacionados con los costes ocasionados tras la realización del proyecto. En primer lugar se llevará a cabo el resumen de los gastos unidos al personal que se verá relacionado con el desarrollo de la aplicación. Los gastos para cada tipo de personal dependerán del cargo que ocupe cada uno de ellos.

Dentro del personal incluido en el desarrollo de la aplicación se ha incluido a Fernando García Albaladejo como el principal desarrollador de la aplicación y a Javier García Guzmán, tutor del proyecto y por lo tanto se computarán las horas de las reuniones y ayudas que se han recibido de su parte.

En la Tabla 55 se observan los gastos ocasionados por el personal.

Personal	Categoría	Coste hombre (€/h)	Dedicación hombre (horas)	Coste (Euros)
García Albaladejo, Fernando	Ingeniero Junior	20	400	8000
García Guzmán, Javier	Ingeniero Senior	36	20	720
Total hombre/hora		56	Total	8720

Tabla 55: Costes personal

Como se puede observar, el coste total asociado al personal es de 12720€.

Es necesario tener en cuenta para los gastos las amortizaciones. Estas amortizaciones estarán presentes tanto en las herramientas de tipo hardware como en las de tipo software.

Para llevar a cabo el cálculo de dichas amortizaciones se utilizarán los siguientes datos:

- A: número de meses desde la fecha de facturación en el equipo es utilizado.
- B: período de vida útil estimado para cada componente.
- C: coste del equipo (sin IVA).
- D: % de uso que se dedica con el componente al proyecto.

Teniendo en cuenta las variables anteriores, el coste imputable se calculará mediante la fórmula de cálculo de amortización:

$$A/B * C * D$$

A continuación se muestran tanto los datos de las herramientas hardware como software con sus respectivas amortizaciones calculadas en la Tabla 56 y la Tabla 57.

Descripción	Coste (Euro)	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de vida útil	Coste imputable
Odenador de Sobremesa	550	100	8	48	91,66
SmartPhone HTC	175	100	8	36	38,88
SmartPhone Huawei	205	100	8	36	45,55
Total					176,09

Tabla 56: Costes herramientas hardware

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de vida útil	Coste imputable
Microsoft Office 2010	139	100	8	12	92,66
Adobe Photoshop 6.0	160	100	6	48	20
Microsoft Office Project 2010	775	100	1	12	64,58
Total					177,24

Tabla 57: Costes herramientas software

Una vez calculados los costes de medios utilizados para el proyecto se procederá al cálculo de otros costes indirectos. Estos costes se pueden observar en la Tabla 58.

Descripción	Precio (mes)	Tiempo de uso (meses)	Coste
Conexión a Internet	45	8	360
Luz	40	8	320
Total			680

Tabla 58: Costes indirectos

Una vez calculados todos los gastos se procede a realizar la suma de todos ellos. Esta suma se puede ver en la Tabla 59.

Costes Totales	
Costes de Personal	8720 €
Costes de Hardware	176,09 €
Costes de Software	177,24 €
Costes Indirectos	680 €
TOTAL	9753,33 €

Tabla 59: Costes totales

Finalmente el coste total del proyecto asciende a 9753,33 euros (nueve mil setecientos cincuenta y tres euros con treinta y tres céntimos).

7 CONCLUSIONES

7	CONCLUSIONES	151
7.1	CONCLUSIONES.....	152
7.1.1	CONCLUSIONES DEL PROYECTO.....	152
7.1.2	CONCLUSIONES DE LA METODOLOGÍA UTILIZADA.....	152
7.1.3	CONCLUSIONES PERSONALES	153
7.2	LÍNEAS FUTURAS.....	154

7.1 CONCLUSIONES

En este capítulo se expondrán las conclusiones obtenidas una vez finalizado el proyecto. Se hará referencia a tres tipos de conclusiones. Por un lado se hablará de conclusiones del proyecto, por otro lado se tratará la metodología utilizada y finalmente se hablará de las conclusiones obtenidas a nivel personal.

7.1.1 CONCLUSIONES DEL PROYECTO

Se ha desarrollado como Trabajo Fin de Grado una aplicación móvil para llevar a cabo los preparativos necesarios para afrontar con éxito una inspección técnica del vehículo. Para el desarrollo del mismo se ha elegido la plataforma Android, puesto que actualmente es la plataforma que cuenta con un mayor número de usuarios y sigue creciendo día a día.

Es una aplicación bastante útil ya que mediante ella se pueden revisar partes del vehículo, que en muchas ocasiones, usuarios poco experimentados se olvidan de mirar antes de llevar a cabo dichas inspecciones, además de ofrecer otras funcionalidades atractivas como la activación de recordatorios o información sobre la documentación necesaria. Todo esto se puede hacer de una forma fácil, rápida y en cualquier lugar mediante un dispositivo móvil y la aplicación instalada en él.

Además es de gran utilidad, puesto que se centra en las inspecciones que debe llevar a cabo un usuario sobre su vehículo, y por lo tanto va ligado a la seguridad de los mismos y por lo tanto a la de los pasajeros que se encuentren en él y a todos los de su alrededor.

Todos los objetivos y funcionalidades que se han explicado en capítulos anteriores han sido realizados con éxito tras la finalización del proyecto, a pesar de verse modificada la planificación inicial, en la que se esperaba acabar a mediados de Junio, pero que por diversos motivos no se pudieron finalizar hasta comienzos del mes de Septiembre.

7.1.2 CONCLUSIONES DE LA METODOLOGÍA UTILIZADA

Como se comentó en capítulos anteriores, la metodología utilizada para el desarrollo del proyecto ha sido una metodología ágil denominada Scrum.

El uso de esta metodología ha facilitado mucho todo el proceso de desarrollo de la aplicación, puesto que ha permitido realizar una solución prácticamente a medida,

de una manera fácil y sencilla, y en la que los requisitos pueden ir variando según se vaya avanzando en el grado de desarrollo de la aplicación.

Por todo ello, se piensa que se ha acertado en la metodología escogida para llevar a cabo dicho proyecto, ya que, se trata de una aplicación basada en una plataforma para dispositivos móviles que continuamente se encuentra actualizándose y cambiando. Por lo tanto, no se aconseja establecer y predecir desde un principio como va a ser la evolución del proyecto, sino que habrá que ir viendo como se va desarrollando el mismo y como a medida que se va avanzando se deben ir modificando los diferentes requisitos para finalizar con una aplicación que cumpla con las expectativas deseadas por el cliente.

7.1.3 CONCLUSIONES PERSONALES

La finalización del presente Trabajo Final de Grado ha supuesto una gran satisfacción a nivel personal, ya que he conseguido terminar un proyecto que considero entretenido y de gran utilidad para situaciones a las que se enfrentan los usuarios en la vida real. Toda esta satisfacción también va unida al empeño, esfuerzo y sacrificio que se ha necesitado para llevar a cabo dicho proyecto y que sin la ayuda recibida por parte de mi tutor Javier no hubiera sido posible.

En el desarrollo de este proyecto se ha puesto en práctica muchos de los conocimientos adquiridos a lo largo de los cuatro años de la carrera, asignaturas como la de Programación Orientada a Objetos en las que se trabajaba especialmente con el lenguaje Java, la de Procesos de Desarrollo del Software centrada en el desarrollo de aplicaciones software o la de Seguridad en Dispositivos Móviles en la cual se tuvo contacto con el desarrollo de aplicaciones para la plataforma Android.

Por otro lado, el desarrollo ha sido algo muy interesante, puesto que en la actualidad este tipo de aplicaciones se encuentran en pleno auge, y hoy en día gran cantidad de la población dispone de un dispositivo en los que se puedan ejecutar dichas aplicaciones y las cuales mejoran en la mayoría de los casos la resolución de problemas a los que se enfrentan los usuarios diariamente.

También hay que destacar que durante el proceso de desarrollo de la aplicación no todo han sido éxitos, puesto que han existido momentos en los que surgían problemas y por los cuales la aplicación se quedaba estancada, tardando horas e incluso días en poder resolverlos, pero que finalmente se han resuelto y han servido para conocer más a fondo el funcionamiento de la plataforma que se ha utilizado.

Además, también cabe destacar que aparte de los conocimientos adquiridos a lo largo de la carrera, se han adquirido otros muchos que se van aprendiendo cuando te enfrentas de manera directa ante un proyecto de este calibre.

Por otro lado, el desarrollo del proyecto me afecta de una forma satisfactoria, ya que de manera indirecta todos los conocimientos adquiridos pueden ser de gran utilidad para el mundo laboral en un futuro cercano, teniendo en cuenta la situación actual del mercado, donde existen millones de terminales móviles y miles de aplicaciones para todos ellos.

7.2 LÍNEAS FUTURAS

Inicialmente se pensó en que la presente aplicación contara con alguna funcionalidad que se consideraba importante, pero que por la limitación de tiempo del que se disponía se decidió no llevarlas a cabo.

Uno de las funcionalidades que se proponen como mejora de la aplicación sería la introducción de un apartado en el que se pudiera calcular la equivalencia de neumáticos para los vehículos. Es decir, mediante los datos que proporcionara el usuario tanto de los neumáticos que lleva instalados en ese preciso momento y los datos que proporciona el fabricante comprobar si dichos neumáticos que lleva montados son correctos y por tanto si podría pasar la inspección de forma favorable con ellos.

Por otro lado, otra funcionalidad que se propone como mejora, sería un apartado nuevo en el que se tratara el tema de los niveles de contaminación de los vehículos. En este apartado se detallaría el tipo de carburante que utiliza el vehículo y se procedería a indicar a los usuarios los niveles máximos permitidos para que fueran conscientes de ello.

Otro cambio que se propone, es añadir más vehículos sobre los que se pueda realizar la revisión, ya que actualmente solo se dispone de las opciones de revisar el coche y la motocicleta.

Otra funcionalidad que se podría añadir consistiría en realizar una búsqueda de los lugares más cercanos donde realizar la inspección del vehículo.

Finalmente, otra funcionalidad que se podría implementar, consistiría en dar la opción al usuario de que pudiera realizar una cita previa directamente desde la aplicación.

BIBLIOGRAFIA

- [1] Google. Android Design [en línea].
< <http://developer.android.com/design/index.html> > [Consulta: 15 Enero 2012].
- [2] SOFTENG. Metodología Scrum [en línea]. < <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html> > [Consulta: 22 julio 2012].
- [3] Android. Installing the SDK [en línea].
<<http://developer.android.com/sdk/installing/index.html>> [Consulta: 20 diciembre 2011].
- [4] Android. Content Providers [en línea].
<<http://edu4android.com/es/guide/topics/providers/content-providers.html>>
[Consulta: 15 marzo 2012]
- [5] Android. Ciclo de vida de una actividad [en línea].
<<http://androideity.com/2011/07/06/ciclo-de-vida-de-una-actividad/>> [Consulta: 15 febrero 2012]
- [6] ITV. Aplicación Android 2011. Disponible en Google Play:
<https://play.google.com/store/apps/details?id=appinventor.ai_itiotajasp.ITV_02&hl=es>
- [7] aCar – Track your vehicles. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=com.zonewalker.acar&hl=es>>
- [8] Wikipedia. Android. [En línea].
<<http://es.wikipedia.org/wiki/Android>> [Consulta: 15 de enero 2012]
- [9] Vogella. Android Animations [En línea].
< <http://www.vogella.com/articles/AndroidAnimation/article.html>> [Consulta: 20 de abril 2012]

APÉNDICES

APÉNDICE A. MANUAL DE USUARIO

Este manual va dirigido a todos los usuarios de la aplicación *MyItv* destinada a dispositivos móviles basados en el sistema operativo Android en su versión 2.3 o superiores.

En la Figura 72 se puede observar el menú principal de la aplicación que aparece cuando se lanza la misma. Como se puede observar, se ofrecen cuatro opciones a los usuarios.



Figura 72: Aplicación final pantalla inicial

En la siguiente imagen Figura 73 se muestra la pantalla tras pulsar la opción “¿Qué revisar?”. Una vez pulsada ésta, se le dará la opción al usuario de elegir el tipo de vehículo que desea revisar.

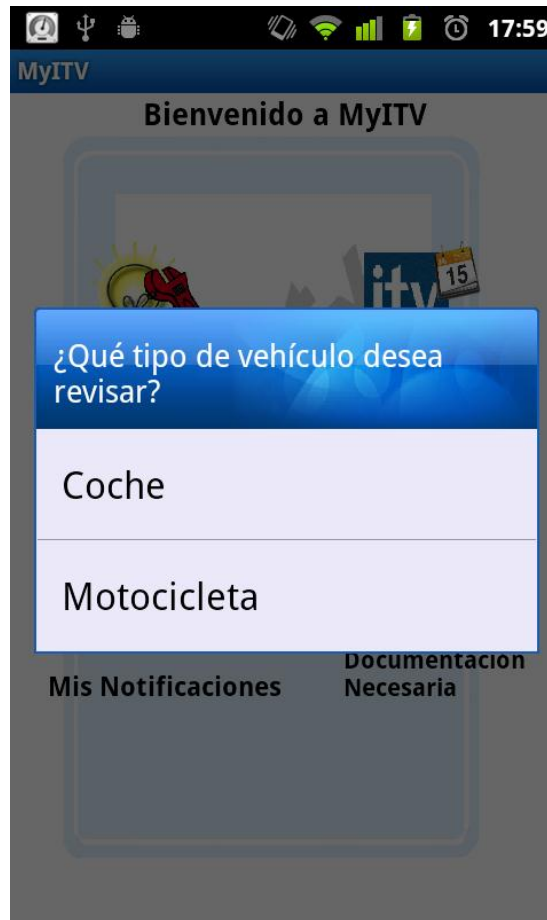


Figura 73: Aplicación final seleccionar tipo vehículo a revisar

Se seleccionará una de las dos opciones, por ejemplo “Coche” y se accederá a la siguiente pantalla Figura 74.

En esta nueva pantalla se pueden observar nuevas opciones que servirán para revisar las diferentes partes del vehículo, en este caso del coche.

También se puede observar que aparece un botón en la barra de acciones en la parte superior izquierda. Esta opción servirá al usuario para poder volver a la pantalla principal de la aplicación en el momento que desee y se mostrará durante todas las pantallas de la aplicación.



Figura 74: Aplicación final que parte revisar coche

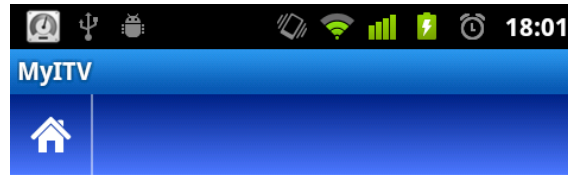
Una vez en esta nueva pantalla se seleccionará una opción, por ejemplo la de “Alumbrado”. Tras elegir dicha opción se accederá a la siguiente pantalla, Figura 75, en la que se mostrará una lista con todas las partes que será necesario revisar.



Figura 75: Aplicación final partes alumbrado coche

En esta lista, como se puede observar, inicialmente aparecen todas las partes marcadas negativamente, y para poder cambiarlo el usuario deberá ir accediendo a cada una de ellas. En este caso se pulsará sobre la opción “Posición”, y se comprobará como debe actuar el usuario para poder cambiarlo.

Una vez pulsada dicha opción se accederá a la siguiente pantalla, Figura 76.



1- Encienda las luces de Posición

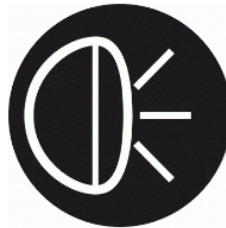
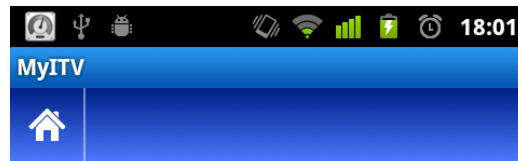


Figura 76: Aplicación final luces de Posición

En esta pantalla se le indicará al usuario el paso que debe realizar y se accederá a la siguiente pantalla mediante la flecha azul situada en la parte inferior derecha de la propia pantalla.

Una vez pulsada dicha flecha se accederá a la siguiente pantalla, Figura 77.



**2- Compruebe que se encienden
todas correctamente**

¿Funciona correctamente?



Figura 77: Aplicación final luces Posición 2

En esta pantalla se le indica otro paso al usuario y se le formula una pregunta a la que deberá contestar, y que dependiendo de dicha respuesta, se modificará la lista de una forma o de otra. En este caso se seleccionará la respuesta afirmativa.

El efecto causado sobre la lista es el que se puede observar en la Figura 78.



Figura 78: Aplicación final lista alumbrado 2

Como se puede observar, el valor del primer elemento ha sido modificado con el nuevo valor seleccionado. El proceso para todos los demás elementos de todas las partes del vehículo es el mismo, incluido para el caso de seleccionar como vehículo la motocicleta, por lo que se decide obviar dichos pasos.

Estando de nuevo en la pantalla principal de la aplicación, se seleccionará la siguiente opción, "Próxima revisión".

Al pulsar esta nueva opción se accederá a la nueva pantalla Figura 79.

Próxima revisión

Seleccione tipo de vehículo

Uso Privado ▼

Fecha matriculación vehículo:

01 sep. 2012

Calcular

Figura 79: Aplicación final próxima revisión

Estando en esta nueva pantalla aparecerá en la barra de acciones un icono de información que servirá al usuario para poder ver los periodos en los que cada vehículo debe realizar las inspecciones de su vehículo, Figura 80.

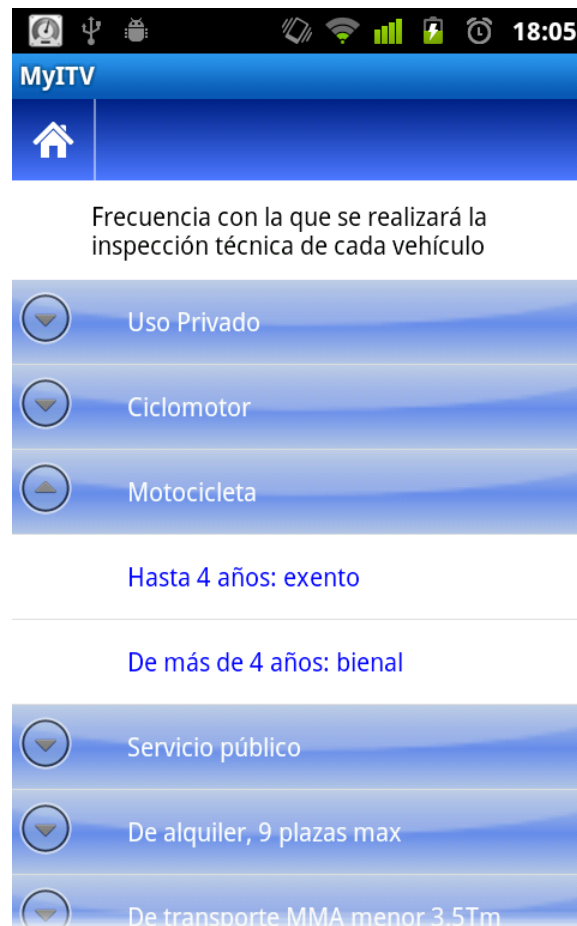
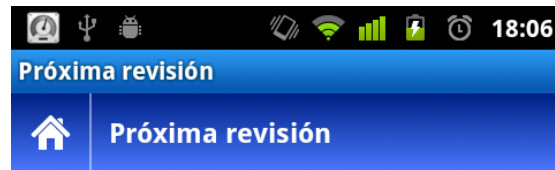


Figura 80: Aplicación final periodo revisión vehículos

Volviendo a la pantalla anterior, Figura 79, también se podrá calcular la fecha correspondiente a la próxima inspección del vehículo. Para ello será necesario introducir el tipo de vehículo sobre el que se desea calcular y la fecha de matriculación del mismo. Una vez introducidos dichos datos se pulsará sobre el botón "Calcular".

Una vez pulsado dicho botón, la fecha para la próxima inspección será calculada, tal y como se puede ver en la Figura 81.



**La fecha de su próxima
inspección es el:**

1-9-2016



Activar notificación para alertar
de la próxima inspección

Figura 81: Aplicación final próxima fecha revisión

Estando en esta pantalla, se le ofrece al usuario la opción de activar una notificación para ser avisado de la inspección. Al pulsar dicha opción se accederá a la siguiente pantalla Figura 82.

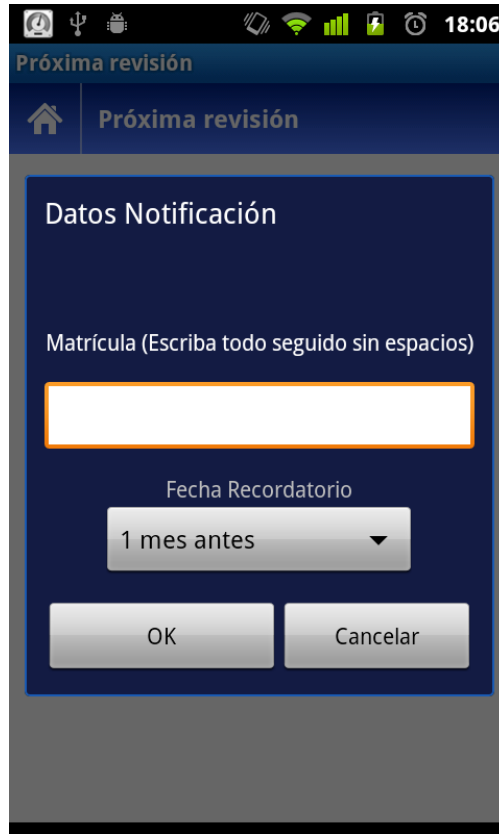


Figura 82: Aplicación final activar notificación

En esta pantalla, se deberá introducir la matrícula del vehículo obligatoriamente, además de hacerlo con el formato correcto, y se elegirá la fecha en la que se desea ser avisado, y para ello se ofrecerán varias fechas como opciones.

Una vez aceptada la notificación, esta será almacenada. Para poder visualizarlas, se deberá volver a la pantalla principal de la aplicación y pulsar sobre la opción "Mis Notificaciones". Una vez elegida dicha opción se podrán ver todas las notificaciones guardadas tal y como se puede observar en la Figura 83.



Figura 83: Aplicación final notificaciones almacenadas

Estando en esta nueva pantalla, se le ofrece al usuario la posibilidad de eliminar las notificaciones que desee. Para ello deberá pulsar sobre el botón de “Eliminar” y se le será mostrada una nueva pantalla, Figura 84.

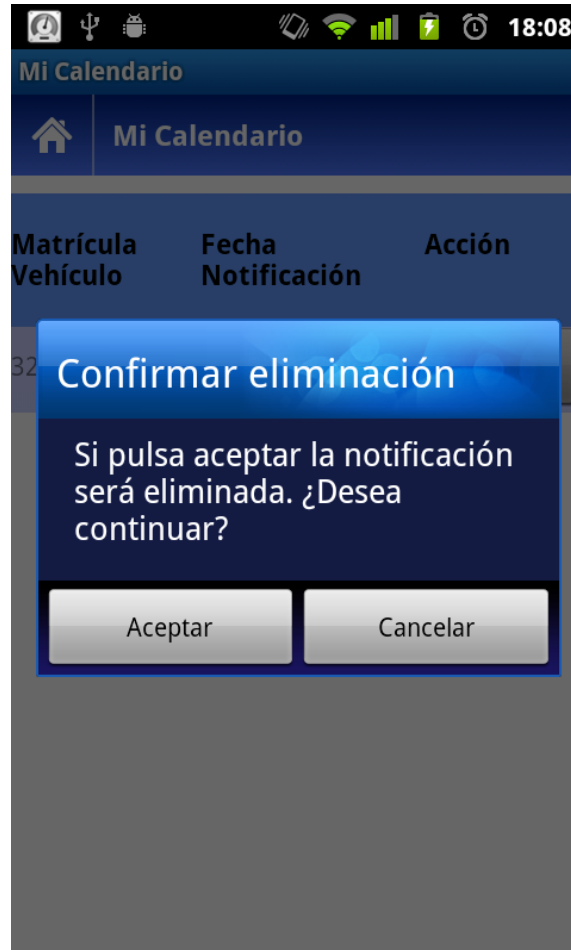
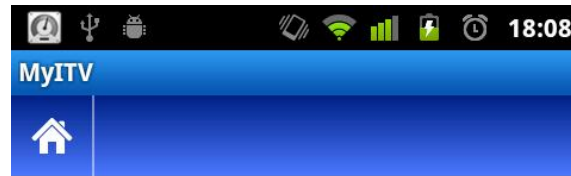


Figura 84: Aplicación final confirmar notificación

Esta pantalla será un simple diálogo de confirmación en el que el usuario deberá confirmar que realmente desea eliminar la notificación.

La última opción a la que se puede acceder desde la pantalla principal de la aplicación es a la de "Documentación Necesaria" y la cual servirá para mostrar al usuario toda la documentación que deberá presentar para realizar la inspección de su vehículo.

Tras elegir la opción comentada se accederá a la siguiente pantalla, Figura 85.



1- Permiso de Circulación del Vehículo



Figura 85: Aplicación final documentación necesaria

En esta pantalla el usuario podrá pulsar sobre la fecha para ver la demás documentación que es necesaria.